UNIVERSITY CARLOS III OF MADRID

Ph.D. Thesis

A user-managed access control model and mechanisms for Web Based Social Networks. Enhancing expressive power, co-ownership management, interoperability and authorized data exposures.

Author:
Lorena González Manzano

Supervisors:

Ana Isabel González-Tablas Ferreres, Ph.D.
Arturo Ribagorda Garnacho, Ph.D.

Computer Science and Engineering Department

Leganés, June 2014

TESIS DOCTORAL

A user-managed access control model and mechanisms for Web Based Social Networks. Enhancing expressive power, co-ownership management, interoperability and authorized data exposures.

Autor:

Lorena González Manzano

Directores:

Dra. Ana Isabel González-Tablas Ferreres

Dr. Arturo Ribagorda Garnacho

Firma del tribunal calificador

Presidente: Juan Manuel Estévez Tapiador

Vocal: Flavio Lombardi

Secretario: María Isabel González Vasco

Calificación:

Leganés, Junio 2014

A José y Mariví, porque sin vosotros no habría llegado este momento.

A Felisa y Alejandro, porque me habéis querido y apoyado como nadie.

A Cristina, porque eres la mitad que me complementa.

A Chema, porque a tu lado comencé y a tu lado terminaré.

Abstract

Web Based Social Networks (WBSNs) are well-known applications which are used by thousands of people worldwide. However, privacy issues, and access control in particular, cannot be disregarded. WBSNs consist of users who upload data to be shared with other users and the management of who is able to access to the uploaded data is a subject to study. In this respect, this thesis focuses on four aspects. First, WBSN users have to specify their privacy preferences in a fine-grained way. Second, WBSN data is not usually related to a single user, who uploads it and who is considered the owner, but to multiple users who are referred to as co-owners. Then, access control has to be managed preserving the privacy of both, owners and co-owners, such that all their privacy preferences are satisfied without restrictions. Thirdly, the great quantity of WBSNs forces users upon being enrolled in many of them, though being access control management a cumbersome task. Lastly, users upload data to WBSNs and providers store it and may use it for unnoticed or unauthorized purposes.

The widespread development of WBSNs has contributed to the enhancement of these applications. The demanding necessity of providing users with tools to control accesses to their data, has boosted the development of proposals in this regard. Nonetheless, a general lack of fine-grained management is detected.

The goal of this thesis is to facilitate fine-grained access control management along the whole usage process within and among different WBSNs in a privacy preserving way. Firstly, an expressive usage control model, together with its administrative model, is proposed to achieve the definition of fine-grained access control preferences.

Based on previous models, a mechanism to manage co-ownership corresponds to the second contribution of this thesis. Data is decomposed in parts and each of them is assigned to the owner or to a co-owner who establishes access control preferences. Then, these preferences are jointly evaluated and the privacy of all

users is completely preserved.

Having the right tools to manage access control in a fine-grained way, the third and last contribution of this thesis is a pair of protocols, one being based on an extension of the other, to attain interoperability, reusability and unauthorized data exposures among different WBSNs. Also taking the proposed usage control model as the underlying base to manage access control, these protocols reduce the burden of managing access control in different applications and thus, they help to increase users' control over their data.

As a result, this thesis aims to be a challenging step towards the enhancement of access control management procedures in the social networking field.

Keywords: Web Based Social Network (WBSN), Fine-grained access control, Access control model expressive-power, Interoperability, Reusability, Data exposure minimization.

Resumen

Las Redes Sociales (RSs) son aplicaciones conocidas y utilizadas a lo largo y ancho del mundo. Sin embargo, los problemas de privacidad, y de control de acceso en particular, no pueden menospreciarse. Las RSs se basan en usuarios que comparten datos entre sí, siendo la gestión de quién puede acceder a dichos datos un tema al que hay que prestar especial interés. En base a esto, la presente Tesis estudia cuatro cuestiones. Primero, los usuarios de las RSs tienen que especificar sus preferencias con alta granularidad. Segundo, los datos de las RSs no se asocian a un único usuario, considerado el propietario y quien sube los datos a las RSs, sino que pueden estar asociados a múltiples usuarios, los cuales reciben el nombre de copropietarios. Por ello, el control de acceso tiene que preservar la privacidad de todos los usuarios, tanto de los propietarios como de los copropietarios, consiguiendo satisfacer las preferencias de control de acceso de todos ellos. Tercero, la gran cantidad de RSs existentes obliga a los usuarios a crear cuentas en cada una de ellas en las que quieran participar, siendo la gestión del control de acceso una tarea tediosa. En último lugar, los usuarios suben sus datos a las RSs y los proveedores de servicio los almacenan, pudiéndolos utilizar para su propio beneficio.

La necesidad de proporcionar a los usuarios las herramientas adecuadas para que puedan controlar sus datos ha acelerado el desarrollo de propuestas para la mejora de las RSs. Sin embargo, se detecta una falta de granularidad en la gestión del control de acceso.

El objetivo de esta Tesis es facilitar la gestión del control de acceso con alta granularidad entre distintas RSs a lo largo de todo el proceso de uso y preservando la privacidad. En primer lugar se propone un modelo de uso expresivo, junto con el modelo administrativo complementario, para conseguir la definición de preferencias de control de acceso con alta granularidad.

Basado en los modelos anteriores, la segunda de las contribuciones se corresponde con el desarrollo de un mecanismo para la gestión de la copropiedad. Los

datos son descompuestos en partes y cada parte asignada al propietario o a un copropietario para que éste establezca las preferencias de privacidad deseadas. Posteriormente, en cada solicitud de acceso a un dato se evalúan todas las preferencias, preservándose así la privacidad de todos los usuarios.

Disponiendo de las herramientas adecuadas para gestionar el control de acceso con alta granularidad, la tercera y última de las contribuciones de esta tesis consiste en el desarrollo de un par de protocolos, uno extendiendo el otro. Estos protocolos facilitan la interoperabilidad, la reusabilidad y la minimización del acceso a los datos de forma no autorizada entre distintas RSs. Igualmente, aplicando el modelo de uso propuesto para la gestión del control de acceso, estos protocolos reducen las tareas a realizar para gestionar el acceso en distintas aplicaciones y por tanto, ayudan a incrementar el control que los usuarios tienen sobre sus datos.

En resumen, esta tesis pretende dar un paso en la mejora del control de acceso en las RSs.

Palabras clave: Red Sociale (RS), Control de acceso granular, Expresividad en modelo de control de acceso, Interoperabilidad, Reusabilidad, Minimización de datos expuestos.

Contents

List of Figures **1**

List of Tables **3**

I Introduction **7**

1 Introduction **9**

1.1 Context 9

1.2 Motivation 12

1.3 Objectives and contributions 15

1.4 Document organization 20

II State of the art **25**

2 Access control in WBSNs **27**

2.1 Access control management in WBSNs 27

2.1.1 Components 28

2.1.2 Enforcement procedure 29

2.2 Access control models for WBSNs 29

2.3 Cryptographic-based approaches to access control in WBSNs 32

2.4 Requirements of user-managed access control in WBSNs 34

2.4.1 Academic proposals and WBSN in use analysis 37

2.5 Expressive power of ACMs for WBSNs 48

2.5.1 Motivation and methodology 50

2.5.2 Specification of access control policies used to evaluate expressive power 52

2.5.3 Analysis of the expressive power of ACMs for WBSNs 53

2.6 Access control administration in collaborative environments. 57
2.6.1 The power of administration 58
2.6.2 Types of administration 59
2.6.3 Analysis of access control administration in collaborative environments 60
2.7 Dealing with co-ownership in access control systems for collaborative environments. 63
2.7.1 Access control management and administration of co-ownership 63
2.7.2 Analysis of co-ownership management and administration in WBSNs 64

3 Interoperability and reusability management. Applications and technologies. 69

3.1 Interoperability definition 70
3.2 Reusability definition 71
3.3 Decentralized social applications 72
3.3.1 Diaspora 72
3.3.2 Friendica 72
3.4 Protocols 73
3.4.1 OpenID 73
3.4.2 OAuth 74
3.4.3 User-Managed Access (UMA) 75
3.5 Files format 77
3.5.1 Friend-Of-A-Friend (FOAF) 77
3.5.2 Microformats 78

III Proposal 79

4 SoNeUCON$_{ABC}$: an expressive usage control model for WBSNs and the enforcement mechanism 81

4.1 Conceptualization of WBSNs 82

4.1.1 Resources 83

4.1.2 Actions 83

4.1.3 Users 83

4.1.4 Relationships 84

4.1.5 Context 85

4.1.6 Summary of the conceptualization 85

4.2 Formalization of SoNeUCON$_{ABC}$ 86

4.3 Access control policies 91

4.3.1 Policy attributes 91

4.3.2 Policy operators 92

4.3.3 Policy construction 93

4.4 Access control enforcement 96

4.5 *SoNeUCON$_{ABC}$* high level architecture 102

4.6 Summary of the chapter 104

5 SoNeUCON$_{ADM}$: administrative model for SoNeUCON$_{ABC}$ 105

5.1 Towards administration 105

5.1.1 Administrative tasks 106

5.1.2 Rights management 106

5.2 *SoNeUCON$_{ADM}$* definition 106

5.2.1 Use rights management 107

5.2.2 Administrative rights management 108

5.3 *SoNeUCON$_{ADM}$* high level architecture 112

5.4 Summary of the chapter 113

6 CooPeD: Co-owned Personal Data management 115

6.1 CooPeD overview 116

6.2 CooPeD description 118

6.2.1 Objects at stake 118

6.2.2 Extension of $SoNeUCON_{ABC}$ usage control model 120

6.2.3 Extension of $SoNeUCON_{ADM}$ 123

6.3 Co-ownership high level architecture 125

6.4 Summary of the chapter 127

7 UMA+FOAF Social Network Protocol. Achieving interoperability and reusability between WBSNs 129

7.1 System overview 130

7.2 System model 133

7.2.1 Architecture 133

7.2.2 Requirements 137

7.2.3 Personal file structure 137

7.2.4 Trust model 138

7.2.5 Adversarial model 139

7.3 U+F protocol description 140

7.3.1 Messages content 140

7.3.2 Execution procedure 143

7.4 Summary of the chapter 150

8 Extended UMA+FOAF Social Network Protocol. Including data exposure minimization and indirect relationships management 151

8.1 System overview 152

8.2 System model 155

8.2.1 Architecture 155

8.2.2 Requirements 157

8.2.3 Trust model 158
8.2.4 Adversarial model 158
8.3 eU+F protocol description 158
8.3.1 Messages content 158
8.3.2 Execution procedure 161
8.4 Data exposure minimization management 164
8.4.1 Traditional PKC 165
8.4.2 IBE-based PKC 166
8.4.3 Comparison: traditional PKC vs IBE-based PKC 167
8.5 Modifying eU+F to fully support $SoNeUCON_{ABC}$: a powerful approach 168
8.5.1 SoNeUCON$_{ABC}$ features management 168
8.5.2 Attributes and policies management 170
8.6 Summary of the chapter 172

IV Evaluation and Conclusions **173**

9 Evaluation **175**

9.1 Evaluation of SoNeUCON$_{ABC}$ 176
9.1.1 Theoretical evaluation 176
9.1.2 Empirical evaluation 181
9.2 Evaluation of SoNeUCON$_{ADM}$ 186
9.3 Evaluation of co-ownership management 189
9.3.1 Policy enforcement for co-ownership management 189
9.3.2 CooPeD prototype 193
9.3.3 Survey study 195
9.4 Evaluation of U+F 201
9.4.1 Theoretical evaluation 201
9.4.2 Experimental evaluation 205

9.5 Evaluation of eU+F 211
9.5.1 Theoretical evaluation 211
9.5.2 Experimental evaluation 217
9.6 Evaluation of U+F vs eU+F 225
9.6.1 Performance analysis comparison 225
9.6.2 Temporal workload comparison 226
9.6.3 Protocols adequacy analysis 228

10 Conclusions **231**
10.1 Conclusions and summary of contributions 231
10.2 Critical analysis on the developed work 234
10.3 Challenges and future research lines 237

V Bibliography and appendices **241**

Bibliography **243**

A Acronyms and abbreviations **265**

B Publications **267**

C Expressive power analysis of ACMs for WBSNs **271**
C.1 Generalizing access control policies 271
C.2 Analysis of the expressive power of ACMs for WBSNs 274
C.2.1 Role based access control (RBAC) models 275
C.2.2 Trust based access control (TBAC) models 277
C.2.3 Relationship based access control (RelBAC) models 282
C.2.4 Attribute based access control (ABAC) models 289
C.2.5 Ontology based access control (OBAC)models 294

D SoNeUCON$_{ABC}$ enforcement functions **299**

D.1 Enforcement functions for $SoNeUCON_{ABC}$ 300

D.2 Enforcement functions for the extension of $SoNeUCON_{ABC}$ 309

List of Figures

1.1 Overview of contributions 19

2.1 Current WBSNs 30
2.2 WBSNs features 35
2.3 Flexible elements in access control policies 36

3.1 Walled garden problem in WBSNs 71
3.2 UMA interactions. Source: [1] 77
3.3 FOAF example 77
3.4 Microformats example 78

4.1 Web Based Social Network Conceptualization 86
4.2 $SoNeUCON_{ABC}$ 87
4.3 Relationships example, $G_{RT[v_8,v_1]}$ 89
4.4 Activity diagram of the enforcement process. 98
4.5 Enforcement process work-flow. 99
4.6 Enforcement process work-flow of the verification of ρ_{rt}. 100
4.7 $SoNeUCON_{ABC}$ high level architecture 103

5.1 $SoNeUCON_{ADM}$ 107
5.2 Administrative objects (AO) management 109
5.3 $SoNeUCON_{ADM}$ high level architecture 113

6.1 Co-ownership management of an object 117
6.2 SoNeUCON coownership management 120
6.3 $T(o_1)$ example 121
6.4 Activity diagram of the enforcement process. 122
6.5 Enforcement process work-flow. 123

6.6 Co-ownership management high level architecture 125
6.7 Hidden techniques 126

7.1 WBSN applying U+F 132
7.2 U+F architecture 134
7.3 Proposed FOAF file including new fields 139
7.4 User logs in to a WBSN 145
7.5 User accesses a to contact FOAF file 148

8.1 U+F vs eU+F 153
8.2 Managed relationships 155
8.3 eU+F architecture 156
8.4 User accesses to the FOAF file of an indirect contact 163
8.5 Traditional PKC - Acquiring User2's resources 166
8.6 eU+F *rt* construction (intermediate nodes omitted for brevity). . . . 169
8.7 eU+F extension to evaluation all kind of access control policies. . . 171
8.8 eU+F adjustments to manage *changes in attributes* 172

9.1 Evaluations overview 176
9.2 CooPeD prototype architecture 195
9.3 Survey study (I) 197
9.4 Survey study (II) 198
9.5 Designed U+F prototype architecture 206
9.6 Alice access Bob's data 206
9.7 U+F temporal workload 208
9.8 U+F temporal workload reused 208
9.9 Temporal workload comparison 210
9.10 Designed eU+F prototype architecture 218
9.11 Temporal costs comparison 220
9.12 General temporal workload 221

9.13 Estimation of temporal workload for indirect relationships 222

9.14 Prototype, Facebook and MySpace total cost comparison 223

9.15 Facebook and MySpace indirect relationships comparison 224

9.16 Comparing U+F vs eU+F 227

List of Tables

1.1 Relationship between problems, objectives and contributions 18

2.1 Analysis of academic proposals related to access control in WBSNs (I) 41

2.2 Analysis of academic proposals related to access control in WBSNs (II) 42

2.3 Analysis of academic proposals related to access control in WBSNs (III) 43

2.4 Analysis of academic proposals related to access control in WBSNs (and IV) 44

2.5 Analysis of access control features in WBSNs in use 47

2.6 Expressive power comparison of ACMs 54

2.7 Administrative features analysis 61

2.8 Co-ownership management analysis 64

7.1 Interchanged messages in U+F 142

8.1 Interchanged messages in eU+F 160

8.2 PKC vs IBE-based PKC 167

9.1 WBSNs structure 181

9.2 Analysis of rt construction 184

9.3 Policies evaluation temporal workload 185

9.4 Policy enforcement temporal workload 186

9.5 Administrative tasks comparison 188

9.6 Explored nodes and TW of rt construction. 190

9.7 Average TW policy enforcement for co-ownership management. Analogous types of policies. 192

9.8 Average TW policy enforcement for co-ownership management. Different types of policies. 192
9.9 Analysis of users' profiles, relative percentages 199
9.10 Analysis of potential users, relative percentages 200
9.11 U+F theoretical evaluation: protocol phases 204
9.12 Analysing the reuse of data in U+F 207
9.13 eU+F theoretical evaluation: protocol phases 213
9.14 eU+F theoretical evaluation: full support of $SoNeUCON_{ABC}$ . . . 216
9.15 eU+F theoretical evaluation: data exposure minimization 217
9.16 Analysing the reuse of data in eU+F 219
9.17 Theoretical comparison U+F vs eU+F 226

10.1 Administrative tasks comparison 234

C.1 Inductive reasoning 272

Part I

Introduction

Chapter 1

Introduction

This Chapter introduces the context of the thesis, the statement of the problem, the main objectives of the thesis, the achieved contributions and the document organization.

1.1 Context

Currently, we are in the era of techno-dependecy and hyper-connectivity [2] and Web Based Social Networks (WBSNs) are remarkable developments in this regard. WBSNs are growing tremendously [3, 4] and since the release of Friendster in 2002 until recent times, many WBSNs, like Facebook, MySpace, LinkedIn, etc., with different purposes but under the same bases, have emerged. Despite the unquestionable benefits of these applications, e.g. the communication of people worldwide, security and specially privacy issues are tough challenges to face up to.

Privacy is defined as "*the condition of not having undocumented personal knowledge about one possessed by other (1983)*" [5]. WBSNs store large amounts of data, some of them is personal and they must be carefully protected and managed, even if it is an issue not thoroughly taken into account by users [6] and, sometimes, confusing [7]. Assorted studies have analysed security and privacy concerns in WBSNs. Specifically, two different perspectives have been considered, from the users and from the researchers point of view. Although users do not consider privacy a primary requirement, researchers promote the creation of systems that look not only after users expectations but after users security as well.

Regarding users perspective, many authors have contributed to the analysis. Becker *et al.* concluded that the total of Facebook users have never used any of the privacy mechanisms provided [8]. Likewise, Acquisti *et al.* analyzed that even Facebook users who are aware of privacy problems continue using it [6]. This matter can be related to the enormous appearance of perceived benefits, as well as to the fact that people may be conscious about internet security but not aware of its threats [9]. By contrast, more recently, some studies reveal that users are becoming much more private [10, 7].

Despite interests and motivations of WBSN users, studied by researchers and highlighted by authorities, privacy is extremely relevant in everybody's life. For instance, in the Universal Declaration of Human Rights, article 12 sets up the right to not have interferences with our privacy[1]. Likewise, according to the Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995, article 8 points out "*Member States shall prohibit the processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life*" [11]. Besides, the United Nations General Assembly has recently adopted "*The right to privacy in the digital age*", such that "*... human rights should prevail irrespective of the medium and therefore need to be protected both offline and online,...*" [12]. In this respect some techniques and mechanisms are applied to allow users to control their data. Indeed, the fact of being private may become an illusion that blinds users and prevents them from identifying what data is really available to the public [7]. Consequently, researchers and practitioners work in the development of data protection measures that help to achieve the same level of privacy found off-line [13].

Considering the aforementioned importance of privacy, together with the increase of WBSNs, a crucial question arises: Do WBSNs provide enough mechanisms to preserve privacy? Although significant developments have been performed like

[1] http://www.un.org/en/index.shtml , last access May 2014

tools that allow users to accept or decline being tagged in a photo, much more work is required [14, 15]. Given that WBSNs are based on managing data of multiple users, access control is a key research area where this thesis contributes.

According to the NIST, "*access control is concerned with determining the allowed activities of legitimate users, mediating every attempt by a user to access a resource in the system*" [16]. In the WBSN field, a pair of issues related to access control are distinguished, one of them associated with achieving protection against WBSN users [17] and another related to achieve protection against WBSN providers [18].

On the one hand, the most common access control developments are based on creating security measures to allow users specify who may access their data. For instance, if certain resources, such as photos, are restricted to friends, access attempts from friends of a friend should be denied. Nonetheless, the key issue is to provide fine-grained access control management, allowing users to specify their preferences in detail, that is, expressive access control is the challenge pursued [19]. For example, some photos, related to the topic "Summer Parties", may be available for friends from June to September 2014 and restricted to friends that are also relatives. Another point to notice is that, traditionally, access control management is performed before data delivery. By contrast, recent developments pose new demands that lead to enforce access control along the whole usage process [20]. For instance, once photos entitled "Party" are accessed, their download may be denied.

Furthermore, as WBSNs manage data of a huge amount of users and some data belong to more users than the owner (that is the user who uploads them to the WBSN), co-ownership access control management is another noticeable issue. For instance, in a photo of a street gang multiple people apart from the owner appear and all of them become its co-owners. Thus, access control should involve the management of owners and co-owners preferences.

Additionally, given the wide variety of WBSNs with different purposes but with similar type of data in use, access control management may become a burden [21]. For instance, the main goal of Facebook is to share photos and comments among friends and friends of a friend. Similarly, but directly focused on meeting people, Badoo allows sharing photos and making comments. Users have to upload and manage access control in all the WBSNs in which they are enrolled, regardless of being the same data at stake in several WBSNs.

On the other hand, WBSN providers may be a threat that requires protection [18]. In most of WBSNs, users can establish which users may access their data but WBSN providers are in full control of uploaded data. Therefore, providers can use data for their own interests without, in many cases, users notice or consent.

1.2 Motivation

The general purpose of this thesis is the enhancement of WBSN access control management, being specially focused on allowing users the specification of fine-grained preferences. The point is to mimic real life [22]. WBSNs researchers have to mimic daily life human interactions and behaviours. In this regard, users should be allowed to interact with any other users regardless of the type and purpose of the WBSNs.

Nevertheless, it should be reminded that privacy is at the top and apart from mimicking real life, the preservation of privacy is the primary requirement [23]. WBSNs can be generally defined as systems where users interact with each other to share data but accesses are restricted to a set of chosen users and forbidden for the rest.

The previous issues lead to the detection of four specific problems addressed in this thesis:

P1. Lack of fine-grained access control systems that give WBSNs users full control over their data.

WBSNs consist of a large number of users who own huge amounts of data and interact with each other through the establishment of relationships. Consequently, the point is to provide procedures that allow users to manage their data in such a way that all possible necessities are satisfied while respecting privacy. Since the development of traditional access control models, generally referred to as Mandatory, Discretionary and Role Based Access Control (MAC, DAC and RBAC respectively), a significant set of models tailored for the WBSN context have arisen. Some models focus on the update and refinement of traditional models, specially RBAC, to be adapted to WBSN demands [24, 25]. On the other hand, new access control models have been specially developed to meet WBSN necessities [26, 27, 28, 29]. Besides, usage control models are noticeable developments as they focus on managing access not only before delivering data but also along the whole usage process [30, 31]. The persistent control of WBSN users' data is a desirable feature and usage control models are key approaches in this regard. Nonetheless, proposed access control models and mechanisms for WBSNs have a pair of deficiencies. First, they are not expressive enough to allow users to specify all their preferences and then, fine-grained access control management is not provided, e.g. specifying the duration of a certain relationship together with the necessity of having a pair of common contacts. Secondly, access control models for WBSNs are not focused on usage control.

P2. Lack of co-ownership management mechanisms which satisfy all user preferences without restrictions.

Commonly, WBSN data belongs to several users, namely, the owner who uploads them and co-owners who are related to them. Thus, access control should manage both, owners and co-owners privacy preferences. Besides, the satisfaction of all user preferences is an essential requirement to prevent violations of some users privacy. Several proposals focus on co-ownership management, being voting schemes the most common solution [32, 33]. However, this type of scheme may violate some users

privacy, e.g. the most voted option scheme violates privacy of WBSN users who do not vote for a concrete option. By contrast, there are other type of works, namely, the one proposed by K. Thomas *et al.* [34] which satisfies all user preferences if a full consensus between owners and co-owners is reached. Nonetheless, this solution is rather limited as, in many cases, no agreement can be found. Therefore, co-ownership access control management should preserve owners and co-owners privacy but in the most flexible way.

P3. Inability to reuse and jointly manage data among different WBSNs.

There are an assorted set of WBSNs with assorted purposes and services and users enrol in all of them that they want to enjoy. For instance, LinkedIn is focused on the professional audience and Facebook on the general public. Due to the lack of interoperability, users have to create as many accounts as WBSNs in which they want to become enrolled. Then, data management is quite laborious because data is stored and managed in each WBSN where they are uploaded. Besides, some data used in a particular WBSN is analogous to the one used in another and there is not possibility of reusing them, e.g. photos in Facebook and photos in MySpace. In this respect, several solutions are proposed [35, 36, 37]. The standard OpenID[2] is an example, which can facilitate identity data interoperability. Other example is User-Managed Access (UMA) protocol [1, 38]. Among other issues, UMA provides users with control over data-sharing, which is a key feature towards resources and access control policies interoperability [1, 38]. However, these contributions do not provide a concrete procedure to achieve interoperability of resusability of resources, identity data and access control policies. Indeed, in practice, there is not a single approach that gets access to data, i.e. photos, or get the reuse of data, i.e. access control policies, of a WBSN to another.

P4. Disclosures of data that lead to the violation of users privacy.

Commonly known, lots of daily news inform about the persistent disclosure of

[2]http://openid.net/ , last access May 2014

WBSN users data, either by service providers[3] or by attackers[4]. WBSN providers are in possession of all uploaded data and, once accepted the "Terms Of Service" at the registration phase, data can be licitly used for different purposes, e.g. commercial issues. On the other hand, service providers, apart from using data licitly (which may violate users privacy), claim that they use security measures that prevent unnoticed data deliveries and disclosures. Nonetheless, multiple attacks have been performed over popular WBSNs[5] and users data has become compromised. Therefore, data has to be appropriately protected against insiders and outsiders, that is, against licit use by service providers and illicit use by attackers. Regardless what the "Terms Of Service" states and attackers could achieve, privacy must be preserved over everything else. To address this matter cryptography has been the primary applied technique in the literature [39, 40]. In general, data is stored encrypted and decryption keys are delivered among authorized users. The main drawback which should be carefully considered is that, specially in systems like WBSNs where many data are accessed by many users, key management is a hard task.

1.3 Objectives and contributions

The general goal of this thesis is to facilitate fine-grained access control management along the whole usage process within and among different WBSNs in a privacy-preserving way.

There was a need to address the previous research topics, which have been reflected in the objectives of this thesis:

O1. Develop a **model to provide expressive** management achieving that **WBSN users have full control over their data**. It should provide a complete

[3] http://www.digitaleyemedia.com/blog/boxes-you-want-to-uncheck-on-linkedin, last access May 2014

[4] http://www.facebook.com/notes/facebook-security/protecting-people-on-facebook/10151249208250766, last access May 2014

[5] http://allfacebook.com/camera-bug-security-loophole-version-1-1-2_b107435, last access May 2014

solution including administrative functions.

O2. Develop a **mechanism to allow co-ownership management** and provide owners and co-owners with the appropriate tools to **jointly manage access control. The privacy of all users has to be preserved**.

O3. Develop a **mechanism that addresses interoperability and reusability** among different WBSNs **reducing the necessity of managing several WBSN accounts**.

O4. Develop a **mechanism to deal with undesirable accesses to WBSN users' data**, **protecting users from privacy violations**. It should ensure the ease of key management.

The achievement of these objectives has led to the next three contributions:

C1. An **expressive usage control model for WBSNs** (see Chapter 4) together with **the complementary administrative model** (see Chapter 5) that allow fine-grained access control management. Depicting a WBSN as a graph, where users are nodes and edges relationships, $SoNeUCON_{ABC}$ and $SoNeUCON_{ADM}$ are proposed. $SoNeUCON_{ABC}$ is an extension of $UCON_{ABC}$ that achieves expressive access control regarding a set of six WBSN features. $UCON_{ABC}$ is an usage control model based on Attribute Based Access Control (ABAC). It is extended in such a way that users, data and relationship attributes are managed. Access control management is carried out in a privacy-preserving way, as apart from the attributes of the administrator and the requester of a particular data, the attributes of the rest of nodes involved in the relationships between them remain hidden. Additionally, $SoNeUCON_{ADM}$ is the administrative model for $SoNeUCON_{ABC}$. It defines the management of revocation, delegation and other administrative tasks. The suitability of both models is evaluated. In $SoNeUCON_{ABC}$ the fulfilment of the required set of WBSN features is theoretically studied. Furthermore, policy enforcement temporal workload is measured through a proof of concept system, concluding the appropriateness of the model for most of the features. On the other hand, the

evaluation of $SoNeUCON_{ADM}$ analyses the satisfaction of administrative tasks.

C2. A co-ownership management mechanism for decomposable objects (see Chapter 6) that allows the preservation of owners and co-owners privacy. WBSN data may be not related to a single user, but to a set of co-owners as well and, for these situations, Co-owned Personal Data management (CooPeD) is proposed. CooPeD presents a novel technique to preserve the privacy of all users satisfying all users demands without restrictions. Inspired by [41], objects are decomposed in parts and assigned to users. Then, each part belongs to a particular owner/ co-owner who individually manages it establishing his access control preferences. CooPeD is developed over $SoNeUCON_{ABC}$ and $SoNeUCON_{ADM}$ by extending both models to integrate co-ownership management. CooPeD has been evaluated by means of a proof of concept system to analyse the feasibility of using $SoNeUCON_{ABC}$ for co-ownership management, a prototype to prove the feasibility of implementing CooPeD, and a survey to study the relevance of co-ownership management and the usefulness and appealing of the proposal.

C3. Mechanisms to achieve interoperability and reusability among different WBSNs including data exposure minimization (see Chapters 7 and 8). A pair of protocols constructed over $SoNeUCON_{ABC}$ are proposed. As a primary step, based on the UMA core protocol [1] and the FOAF project [42], UMA+FOAF Social Nework protocol (U+F) is proposed. It attains interoperability and reusability of resources, identity data and access control policies across different WBSNs, where resources mainly correspond to photos, videos and audio files and identity data refers to the users' profile and contacts data. This protocol exclusively focuses on direct relationships. Next, extended U+F Social Network protocol (eU+F) is proposed, including the protection of data against WBSNs providers and the management of indirect relationships, being essential the latter feature to allow fine-grained access control. Indirect relationships facilitate the management of WBSN features like multi-paths, which allow accessing users' data through mul-

Problem	Objective	Contribution
P1: Lack of fine-grained access control systems that give WBSNs users full control over their data.	O1: A model to provide expressive management, together with administrative functions.	C1: An expressive usage control model for WBSNs and its administrative model.
P2: Lack of co-ownership management mechanisms which satisfy all user preferences without restrictions.	O2: A mechanism to allow co-ownership management satisfying all owners and co-owners privacy preferences.	C2: A mechanism to manage co-ownership for decomposable objects in WBSNs.
P3: Inability to reuse and jointly manage data among different WBSNs.	O3: Interoperability and reusability between WBSNs.	C3: A mechanism to reach interoperability and reusability among different WBSNs that also minimizes unauthorized data exposures.
P4: Disclosures of data that lead to the violation of users privacy.	O4: A mechanism to deal with undesirable accesses to WBSN users' data, protecting users from privacy violations.	

Table 1.1: Relationship between problems, objectives and contributions

tiple user relationships [17]. A prototype is developed to verify the feasibility of implementing both protocols in a simulated environment, as well as to compare their workload regarding a pair of well-known WBSNs, Facebook and MySpace. Moreover, U+F and eU+F are jointly compared. As a result, it is noticed that the temporal workload of eU+F, as expected, is higher than that of U+F. Besides, results show that both protocols can be considered acceptable and challenging approaches that, even supposing a workload increase in comparison with Facebook and MySpace, satisfactorily attain all established requirements.

The relationship between the detected problems, the objectives and the achieved contributions is shown in Table 1.1.

We find that these issues are a step towards the enhancement of current WBSNs access control management procedures. All contributions of this thesis are summarized in Figure 1.1. The process starts by uploading and managing data in WBSNs,

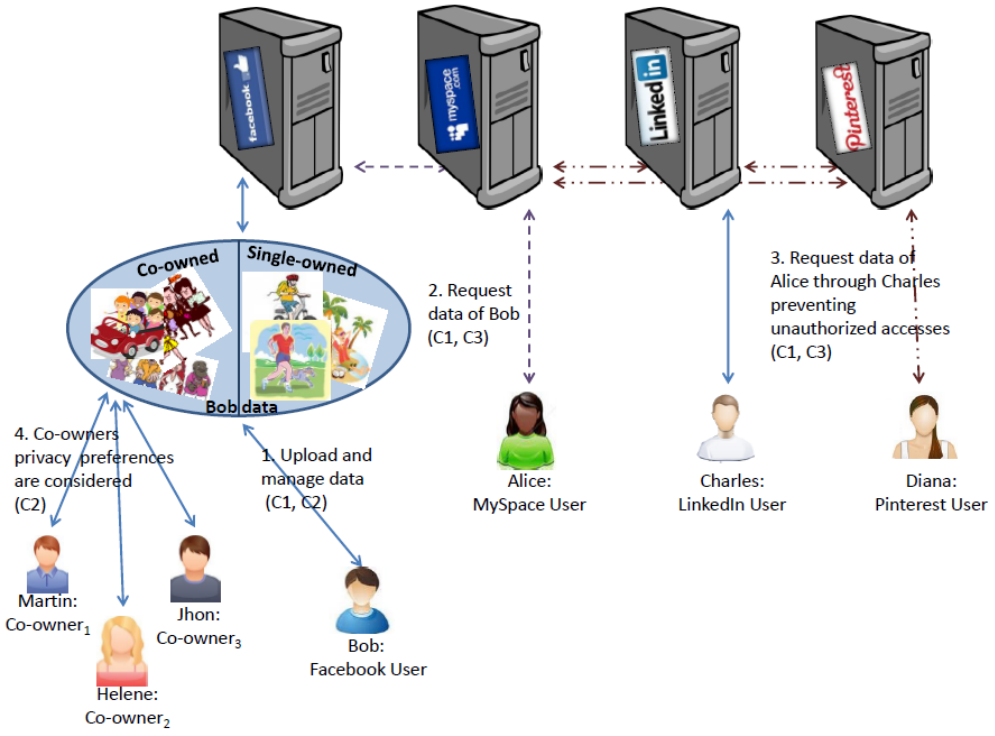


Figure 1.1: Overview of contributions

together with the establishment of access control policies (Message 1 of Figure 1.1). Then, users are able to request data of users enrolled in any other WBSN, including those whose access is direct (msg. 2 of Figure 1.1) or those whose access is indirect and has to be performed through other user/s (msg. 3 of Figure 1.1). Besides, unauthorized accesses to data are prevented applying cryptography and managing access along the whole usage process (msg. 3 of Figure 1.1). Furthermore, note that in case of co-owned data, access control enforcement involves the evaluation and the satisfaction of the owner's and co-owners' policies (msg. 4 of Figure 1.1).

Nonetheless, privacy problems caused by the use of external devices, such as photo cameras, or other elements like systems to record on-screen activities, are left out of the scope of this thesis. On the other hand, users with whom a relationship is established, are always considered trusted. Thus, the identification of unexpected malicious changes in their behaviour is a matter of future work. In case users

behaviours differ from the common one, e.g. using an anonymity server to preserve their identity, they would be considered untrusted WBSN users, being tools to monitor trustworthiness a starting point of research [43].

The research results published in scientific journals and conferences during the development of the present thesis are listed in Appendix B.

1.4 Document organization

This thesis is composed by ten chapters organized in five parts:

Part I. Introduction. This part introduces the document and it includes this Chapter.

Chapter 1. Introduction. This Chapter involves the context of the thesis, the statement of the problem, the research objectives and the proposed contributions.

Part II. State of the art. This part analyses the state of the art associated with this thesis. The analysis has been organised into a pair of chapters.

Chapter 2. Access control in WBSNs. This chapter presents an overview of current trends in WBSN access control management. It introduces the main concepts of access control management to afterwards, present access control models for the WBSN field and specify WBSN access control requirements. Moreover, access control administration and co-ownership management in collaborative environments are analysed.

Chapter 3. Interoperability and reusability management. Applications and technologies. This chapter introduces applications and technologies currently applied for interoperability purposes.

Part III. Proposal. This part contains the set of developed proposals to satisfied the objectives of this thesis. There are a total of three contribution, the first of them is divided in Chapters 4 and 5, the second proposal is presented in Chapter 6 and the third and last contribution is split into Chapters 7 and 8.

Chapter 4. SoNeUCON$_{ABC}$: an expressive usage control model for

WBSNs. This Chapter presents $SoNeUCON_{ABC}$. It extends $UCON_{ABC}$ [44] including relationship management, allowing privacy-preserving and fine-grained access control management. Subjects, objects and relationships attributes are managed in such a way that, apart from the attributes of the administrator (the owner) and the requester of a particular data, the attributes of the rest of users remain hidden. Besides, expressive power is attained by the definition of an access control policy language along with its enforcement procedures.

Chapter 5. SoNeUCON$_{ADM}$: an administrative model for SoNeUCON$_{ABC}$. This Chapter introduces $SoNeUCON_{ADM}$, an administrative model for $SoNeUCON_{ABC}$. Basically, it presents the management of use and administrative rights which involve managing administrative objects, delegation and revocation.

Chapter 6. CooPeD: Co-owned personal data management. In this Chapter a co-ownership management mechanism is presented. It protects privacy of all WBSN users to whom an object is related. The proposal is applicable to decomposable objects. It consists of managing objects as they where composed of parts. Each part belongs to a particular user who individually manages it. Besides, it is developed over $SoNeUCON_{ABC}$ and $SoNeUCON_{ADM}$.

Chapter 7. UMA+FOAF Social Network Protocol. Achieving interoperability and reusability between WBSNs. This Chapter describes UMA+FOAF Social Network Protocol (U+F) to achieve interoperability and reusability of identity data, resources and access control policies among different WBSNs. It focuses on UMA protocol [1, 38] and the FOAF project [42]. Furthermore, it works on top of $SoNeUCON_{ABC}$ and access control management is performed accordingly.

Chapter 8. Extended UMA+FOAF Social Network Protocol. Including data exposure minimization and indirect relationships management. In this Chapter U+F is extended in two ways. First, eU+F includes indirect re-

lationships management to achieve fine-grained access control. Second, a hybrid cryptographic approach is applied to protect data against unnoticed deliveries. Moreover, this proposal works over $SoNeUCON_{ABC}$.

Part IV. Evaluation and Conclusions. In this part the evaluation and the conclusions of proposed contributions are presented in two different Chapters.

Chapter 9. Evaluation. The evaluation of the contributions of this thesis is presented in this Chapter. The evaluation involves:

- $SoNeUCON_{ABC}$ is evaluated analysing its expressive power and testing the feasibility of the proposal through a proof of concept which calculates the temporal workload of the enforcement process.

- $SoNeUCON_{ADM}$ is analysed assessing the completeness of the model through a comparison with a pair of administrative models.

- CooPeD is evaluated, in first place, to assert the feasibility of its application over $SoNeUCON_{ABC}$. Secondly, a prototype helps to analyse the possibility of implementing CooPeD. Lastly, a total of 206 people worldwide have been surveyed to test the usefulness and the relevance of the proposal.

- U+F is theoretically evaluated discussing the satisfaction of established requirements and analysing its performance. Also, the performance and the applicability of U+F is empirically analysed by a prototype development. Additionally, results are compared with two challenging WBSNs, Facebook and MySpace.

- eU+F is also theoretically and empirically analysed. The fulfilment of proposed requirements and its performance is theoretically studied. By contrast, enhancing the prototype developed for the evaluation of U+F, the applicability of eU+F is empirically analysed. The protocol is compared with Facebook and MySpace as well.

- U+F and eU+F are jointly compared, theoretically, to analyse applied elements and empirically, to study the temporal workload of every protocol's phase.

Chapter 10. Conclusions. In this Chapter the conclusions of this thesis are described. It involves a critical discussion of the work performed and the outline of future and open research issues.

Part V. Bibliography and Appendices. This part includes the bibliography in use, the scientific publications derived from the underlying research, and a set of appendices that complement the main content.

Bibliography. The list of references to other research papers, technical documents and standards used in the thesis are presented herein.

Acronyms and abbreviations The set of acronyms and abbreviations that are used throughout this thesis are described herein.

Publications. The papers related to this thesis in which the author has participated are listed herein.

Part II

State of the art

Chapter 2

Access control in WBSNs

The huge increase of WBSNs applications promotes the persistent development of new technological advances. In regard to the great amount of data and users involved in WBSNs, access control plays a key role. Users have to control, at every moment, who and under which circumstances their data remains accessible.

This chapter presents an overview of current trends in access control management for WBSNs. First, Section 2.1 presents the main concepts of access control management. Next, access control models for WBSNs are described in Section 2.2. In Section 2.3 the use of cryptography to provide access control is presented. Afterwards, requirements regarding access control in WBSNs are defined in Section 2.4. Section 2.5 depicts an study of the expressive power of access control policy languages. In Section 2.6 access control administration in collaborative environments is described. Finally, co-ownership management in access control systems for collaborative environments is presented in Section 2.7.

2.1 Access control management in WBSNs

Access control is the process of mediating requests of data managed by a system and determining whether the request is granted or denied. In this regard, access control policies, models and mechanisms are defined [45]. Access control policies are generally described as high-level rules which regulate who/ under which constraints access is granted. Related to policies, access control models provide a formal representation of policies, as well as the way they are enforced. Likewise, access

control mechanisms correspond to the implementation of models, being noticeable that a particular model can be implemented by different mechanisms. For instance, discretionary policies are formalized in the access matrix model. This model is represented as a matrix where rows are subjects and columns are objects of the system. Then, for each request (subject, object), the access matrix determines which actions are available. Besides, this model can be implemented in different mechanisms, for instance, applying an access control list where each object is associated with a list of subjects and a list of the actions that can be performed over objects [46].

Concerning the WBSN field, this Section presents components involved in access control management (Section 2.1.1) and the general access control enforcement procedure (Section 2.1.2).

2.1.1 Components

Access control is one of the most relevant services related to data management systems such as WBSNs, in which a huge quantity of users and data are at stake. Its main goal is to protect resources and identity data from unauthorized users and/or applications. In order to achieve this issue, the following components take part in its provision [47]:

- **Users**: according to the system different sets of users are identified. In particular, in the WBSN context, for simplicity, users are usually classified as administrators, who are the data owners and responsible of establishing access control policies, and requesters, who request access to resources and identity data. Nonetheless, proposals such as [48, 32] distinguish five types of users: *disseminators*, who share owned data with other users; *stakeholders*, who are users tagged within other users' data; *contributors*, who publish data in other users' space; *owners*, who own data; and *accessors*, who request access to data.

- **Identity data**: it refers to users' profiles and contacts' information.

- **Resources**: it refers to photos, videos, comments and any other objects, distinct from identity data, managed in WBSNs.

- **Access control policies**: they refer to high-level rules related to resources and identity data.

- **Reference monitor**: it is the core component of access control management architectures. It consists of two elements, the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP). They are standardized with the X.812 access control framework (ITU-T, 1995), being also known as Access Control Decision Function (ADF) and the Access Control Enforcement Function (AEF) [49]. The former provides affirmative or negative responses in regard to the requested rights on a particular data according to defined policies. The latter enforces decisions taken by the PDP. Both elements are considered trusted entities.

2.1.2 Enforcement procedure

Access control mechanisms determine the concrete implementation of access control enforcement procedures. In general, users request data to the reference monitor. Then, the reference monitor identifies access control policies to satisfy and, regarding their evaluation, the access is granted or denied.

Focusing on existing WBSNs, each of them stores uploaded resources, specified identity data and established access control policies (see Figure 2.1). Thus, regardless of applied policies, models and architectures, access control enforcement is carried out in the reference monitor each WBSN possesses.

2.2 Access control models for WBSNs

The emergence of computer systems supports the development of access control models (ACMs). Traditional ACMs can be classified as Mandatory Access Control

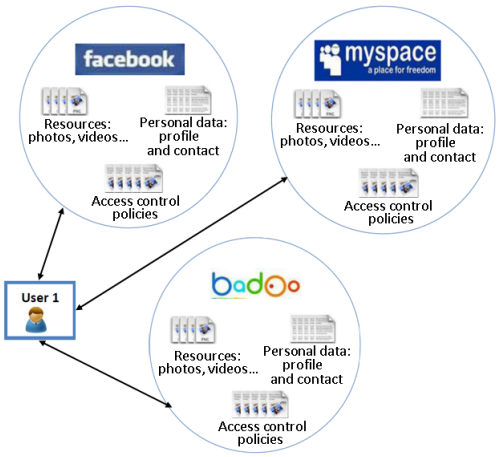


Figure 2.1: Current WBSNs

Model (MAC) where objects and subjects are classified according to security levels and access is granted in regard to them; Discretionary Access Control (DAC), in which access to information is carried out in respect to the user's identity and a set of authorizations or rules; and Role-Based Access Control (RBAC) which focuses on the definition of different roles, assigning permissions (rights) to roles, and, then, assigning roles to subjects. Additionally, Attributed-Based Access Control Model (ABAC) is a rather recent model that is currently receiving a lot of attention. Access is granted or denied in regard to subject, data and environment attributes [50, 51, 52, 53]. Nevertheless, current software developments in which service providers and huge amount of users and data become involved, promote the evolution of traditional ACMs and the development of new ones.

Regarding WBSNs, RBAC is the traditional approach that has received more attention. It has been extended in many ways to meet WBSN requirements [24, 25, 54]. Administrators assign roles to chosen WBSN users to mean that a relationship is established among them. Access control is managed through roles and it can be compared with the establishment and management of groups.

The ABAC model has been also applied to the WBSN context. C.W.D. Munckhof proposes an ACM that automatically selects a particular policy for a post or an added message based on its attributes [55]. On the other hand, J. Park *et al.*

present ACON, an ACM based on managing user, data and session attributes [30].

Nonetheless, some other authors propose models specially developed for the WBSN context. Several works present access control models that can be classified as Relationship-Based Access Control models (RelBAC) [26, 27, 28, 32]. WBSNs deal with an assorted kind of relationships like unidirectional, bidirectional, direct, indirect, etc., and the RelBAC models provide appropriate management procedures that address these aspects. The first RelBAC model was presented by F. Giunchiglia *et al.* and the novelty of this approach lays in the management of permissions as binary relationships between users and data [26]. Similarly, other RelBAC models focus on capturing the essence of social relationships through the management of relationships between pairs of users, usually, administrators and requesters [27, 28, 32].

Also associated to WBSNs, Trust-Based Access Control Models (TBAC) have been developed. They focus on using trust as a condition to access data. Depending on the level of trust of users and data, access is granted or denied. For instance, B. Carminati *et al.*'s proposal focuses on managing the type, depth and trust of relationships between pairs of users [28]. Another related contribution is Personal Data Access Control (PDAC), which focuses on trust computations to identify the users that may access data [56]. Trust in this case is computed through the level of trust that some users, who are authorized, put in others. A simpler approach is developed in [57]. Different levels of trust are assigned to objects and access is granted to users who prove having an equal or a higher level of trust than the requested object.

Finally, another type of WBSN ACMs is noticed, Ontology-Based Access Control (OBAC) models. They consider assorted and fine-grained access control management systems based on the use of ontologies specially developed for WBSNs [58, 59, 29].

2.3 Cryptographic-based approaches to access control in WBSNs

Several approaches combine cryptographic techniques with access control management to provide data confidentiality. Five different categories of access control cryptographic approaches can be distinguished in the WBSN area.

The first category involves the most simple cryptographic approaches. Basically, data is encrypted with an encryption key and decryption keys are distributed among chosen users [60, 61, 62, 40, 63, 64, 65]. The main drawback of this kind of approaches is key management complexity because decryption keys have to be distributed to all appropriate users.

Y. Zhu *et al.* and L. A. Cutillo *et al.* propose group-based cryptographic proposals ([66] and [67] respectively). The general idea behind these works is that groups agree upon encryption keys and the decryption ones are shared between every group member. Nonetheless, key distribution continues being a hard task.

Trying to alleviate key management, K. B. Frikken *et al.*'s proposal lays the bases of the third category. Derivable keys in relation to the distance between WBSN users are used in encryption and decryption processes. Despite the decrease of keys distribution complexity, derivable keys may cause that undesirable users could decrypt certain data [68].

The fourth category focuses on hybrid cryptographic schemes in which symmetric and asymmetric cryptography is jointly applied. In particular, K. Graffi *et al.*'s work falls in this category. They propose an hybrid scheme where data is symmetrically encrypted and symmetric keys are asymmetrically encrypted for each authorized user [39]. Thus, despite the efficiency of the proposal concerning keys distribution, storage space is neglected. Symmetric keys per encrypted data have to be encrypted and stored for all chosen users.

From a different point of view and also avoiding key management problems, Attribute Based Encryption (ABE) cryptography and Identity Based Encryption

(IBE) cryptography establish the fifth and last category. On the one hand, ABE focuses on creating a pair of keys, to encrypt and decrypt, in regard to an established group of attributes. ABE schemes can be divided into two groups, Ciphertext-Policy ABE (CP-ABE) and Key-Policy ABE (KP-ABE). The former corresponds to the association of policies with ciphertexts and attributes to users' keys and it is a remarkable technique in applications in which data is managed by multiple profiles, such as in hospitals or in the army [69]. By contrast, the latter corresponds to the attachment of policies to user keys and attributes to ciphertexts, being useful in applications like auditing logs [51]. The main difference between both approaches is that in CP-ABE attributes of users keys are known, while in KP-ABE they are hidden. Particularly, four proposals apply ABE in such a way that WBSN data is encrypted applying attribute predicates and users with the right attribute keys can decrypt them [70, 71, 72, 73].

On the other hand, the primary innovation of IBE is the use of user identity attributes, e.g. email, address and so on, for encryption and decryption operations [74]. Cryptographic keys are created from public parameters together with chosen user identity attributes. In particular, IBE is applied in R. Schlegel *et al.*'s work [75]. Either ABE and IBE have meaningful advantages, namely, the unnecessary use of a public key infrastructure which avoids certificate management and reduces the system complexity and the cost for managing keys. However, both approaches have some common drawbacks. First, they are particularly costly in terms of computation [76]. Second, though depending on particular implementations, attribute certificate authorities to assert the validity of attributes come into play. Lastly, the key escrow problem [77], defined as the existence of third parties which may access decryption keys, is inherent in IBE and ABE approaches.

In view of the foregoing, cryptography inherently increases access control management complexity, being specially affected by keys management. Several approaches release this tedious process (fourth and fifth category) but they may cause

problems of storage space or the necessity of additional entities. Therefore, the best option depends on concrete systems' requirements.

2.4 Requirements of user-managed access control in WBSNs

Access control allows users to grant or deny a certain right over particular data. However, not all access control systems pursue the same objectives, e.g. an educational application has to look for parental access control management and, by contrast, a business application requires managing access control in existing departments. Focusing on WBSNs, in 2006, C. Gates proposed a set of requirements in order to provide user-managed access control in the Web 2.0 [78]:

- *Relationship-based*: data administrators control the release of data based on the established relationships with data requesters, instead of delivering information depending on the requester role or any other feature.

- *Fine-grained*: users must control their information in a fine-grained way, choosing who is able to access it and under which circumstances. It should be possible to define fine-grained policies for both data, requesters and relationships.

- *Interoperability*: users access multiple WBSNs and want their data to be used in a similar way in many of them. Access control systems should be interoperable between different WBSNs, so, it would be possible that user preferences follow the user whatever WBSN is used.

  Furthermore, interoperability should be referred to any type of element managed in WBSNs, that is, identity data, resources and access control policies. On the one hand, WBSN data can be classified as identity data and resources. On the other hand, access control management involves access control policies which are considered WBSN elements.

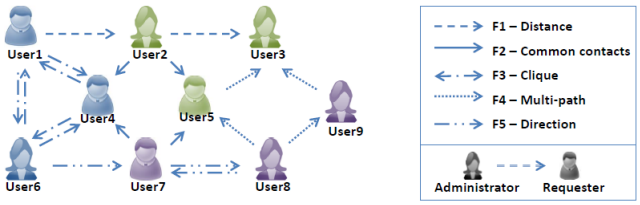


Figure 2.2: WBSNs features

- *Sticky policies*: policies should follow the data to which they apply, preventing from uncontrolled data disclosures after being released, that is, policies should be attached to data. This issue is related to usage control. Access control should be enforced throughout the whole usage process [44].

Nonetheless, several proposals and recent developments help to the identification of a new requirement which is considered of special importance:

- *Protection against honest-but-curious servers* [18]: WBSN users' data has to be protected against WBSN providers. Indeed, WBSN providers control all uploaded data and they can be used without being appropriately notified to WBSN users.

On the other hand, alluding to the *fine-grained* requirement, some features are identified. The satisfaction of this requirement requires ACMs to have expressive power, that is, the ability to express a wide range of policies [79]. Specifically, in the social networking field and according to literature the consideration of the following set of features, depicted in Figures 2.2 and 2.3, facilitates the development of expressive WBSNs ACMs:

F1 *Distance* [80, 28]: WBSNs are composed of a vast quantity of users who interact between each other. However, two users of a WBSN may not be directly connected but indirectly, that is, a direct relationship does not exist between them but a path connecting both users can be found considering other users and their relationships. For instance, depicted in Figure 2.2, User3 is indirectly connected to User1 through User2.

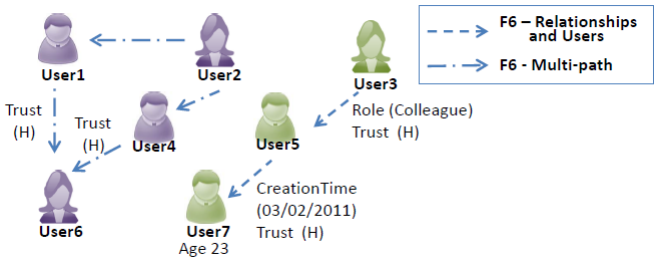


Figure 2.3: Flexible elements in access control policies

F2 *Common-contacts* [80, 19, 81]: WBSNs users have, in multiple circumstances, common contacts. In this proposal such contacts are defined as the establishment of an unidirectional relationship (see Feature 5) from the administrator to his contacts and an unidirectional relationship (F5) between the requester and his contacts. For instance, in Figure 2.2 User2 is unidirectionally connected with User4 and User5, and User7 has an unidirectional relationship with both users too. Then, User2 and User7 have User4 and User5 as common contacts.

F3 *Clique* [80, 19]: a set of WBSNs users forms a clique, that is they belong to a close-knit group in which all contacts are bidirectionally connected between each other. For instance, User1, User4 and User6 depicted in Figure 2.2 form a clique.

F4 *Multi-path* [82, 17]: WBSNs consist of users that establish connections with other users of the WBSN. When two users are not directly connected but indirectly, it is said that a path exists between both. In particular, several paths may exist between two users, that is, with each path involving a different set of ordered nodes. For instance, in Figure 1, User3 is connected to User8 by a pair of paths which are different because they are composed of a distinct sets of nodes.

F5 *Direction* [28, 19, 83]: a relationship can be established in an unidirectional or bidirectional way. The former corresponds to the case in which a relationship

request is only established in one direction. For example, in Figure 2.2, User6 has acknowledged that User7 has a relationship with herself but User7 has not acknowledge any relationship between User6 and himself. On the contrary, the latter, a bidirectional relationship, implies that both users have acknowledge the existence of a relationship between the other and him/ herself.

F6 *Flexible elements in access control policies* [83, 84]: WBSNs involve users, resources, as well as identity data, and relationships. Then, access control policies should be defined applying any characteristic related to these elements, i.e. the users' age, as well as the combination of characteristics, e.g. the users' hobbies and the relationships' expiration date. For instance, in Figure 2.3, User3 grants access to users (e.g. User7) who are over 20 and under 30 (20 < age < 30) and who are distanced a pair of hops (path with length two), where the relationship of the first hop is highly (H) trusted and has the role colleague and the relationship of the second hop is highly trusted and started before 2012. Also regarding fine-grained management, in Figure 2.3, User2 establishes that given a particular requester (such as User6), all users involved in every path that connects the requester and User2, have to highly trust the requester.

2.4.1 Academic proposals and WBSN in use analysis

In the light of WBSN requirements, their management poses appealing issues at stake. In order to analyse recent advances in user-managed access control systems for WBSNs, a set of 50 academic proposals that work on privacy in WBSNs and 9 of the most currently used and active WBSNs are assessed against the requirements previously described.

Tables 2.1-2.4 and 2.5 summarize the results of the analysis. In general terms, academic proposals largely differ from WBSNs in use. Academic contributions are specially focused on cryptographic access control procedures and the development of

novel techniques to facilitate access control management. By contrast, the majority of active WBSNs tend to manage access control in a simpler way, neglecting important requirements such as *interoperability* or *protection against honest-but-curious servers*.

In order to structure the analysis, in each proposal and WBSN in use, the following features related to aforementioned requirements have been examined:

- *Relationship-based*
  - All academic proposals and WBSNs in use are based on relationships and then, this feature is not studied.
- *Fine-grained*
  - **Elements in access control policies**: specification of elements involved in access control policies. Note that the management of indirect relationships, which does not require any particular element but a specific algorithm or mechanism exists, is pointed out with symbol γ.
  - **Policy definition**: specification of languages or tools used to develop policies, like XML.
  - **Details on policy evaluation and enforcement procedures**: description of ways to carry out policy evaluation and specific enforcement procedures.
- *Interoperability*
  - **Elements to interoperability**: specification of elements applied towards the development of interoperable systems.
  - **Interoperable elements**: specification of elements managed in WBSNs (identity data, resources and access control policies) that could be interoperable between different WBSNs.

- *Sticky policies*
  - **Sticky elements**: specification of elements used to implement sticky policies.
- *Protection against honest-but-curious servers*
  - **Protection techniques**: indication of techniques used to prevent servers from accessing data that users do not authorize.
  - **Protection elements**: specification of the type and quantity of elements used to protect data. For instance, a pair of public/private keys per user.
  - **Trusted entity**: identification of trusted entities. They can be users, groups of users or a particular device, for instance, a storage device.

Finally, there are a pair of symbols to notice. Symbol "X" refers to the fact that a given approach has explicitly mentioned that an aspect is out of the scope of the proposal, and symbol "-" implies that an approach does not point out anything about a specific issue.

2.4.1.1 Academic proposals for WBSNs

A total of 50 academic proposals focus on access control in WBSNs. The summary of the analysis is presented in Tables 2.1, 2.2, 2.3 and 2.4.

Regarding the *fine-grained* requirement the analysis draws interesting results. It is noticed that 16 approaches manage roles [85, 86, 35, 36, 37, 87, 63, 70, 71, 72, 65, 88, 73, 24, 54, 89] and 9 works also manage trust [90, 58, 57, 67, 65, 31, 91, 56, 28]. By contrast, few approaches manage assorted elements such as users' location [36], data attributes [92, 73, 93, 30, 55, 29], users attributes [85, 88, 94, 95, 96, 93, 30, 29] or time periods [88]. Furthermore, extremely similar to current WBSNs, a total of 12 approaches manage relationship types, that is, being friend of, worker of, etc [28, 94, 91, 27, 97, 83, 32, 95, 96, 98, 99, 58]. On the contrary, assorted

elements are supported by [93, 30], mainly, conditions and obligations. The former set corresponds to environmental or system factors considered in the access control enforcement process and the latter set refers to elements to satisfy before, during of after the access is granted. Moreover, relationships distance is also considered and several works support and manage indirect relationships [90, 58, 70, 65, 73, 28, 94, 56, 91, 97, 83, 95, 96]. Related to relationships, a noticeable feature common to most of proposals is the management of unidirectional relationships. This matter reflects that access control mechanisms are established from one user to another and the particular implementations are the ones which make relationships bidirectional.

In what concerns *interoperability*, 12 contributions address this requirement [57, 92, 100, 86, 35, 36, 37, 61, 60, 88, 101, 1]. In general, resource interoperability demand requesters to obtain elements to get access to data. These elements can be referred to as tokens. Particularly, the ITU-T X.812 recommendation, which focuses on data networks and open systems communication security, describes tokens as elements possibly created by requesters and composed of multiple information [49]. Besides, the recommendation highlights that tokens differ from certificates in that certificates are delivered by a trusted authority. In this study only [88] and [1] implicitly mention the use of tokens, in particular, OAuth tokens[1]. However, in the remaining works that address interoperability, tokens are represented in multiple ways, such as XML files [100] or tickets [92], and can be composed of assorted features, like keys [100] or digital signatures [86]. One relevant common point of some proposals is the use of tokens to attest relationships [35, 86, 57, 100, 60].

Conversely, identity data *interoperability* has not received a lot of attention. Some approaches focus on the interoperability of users' profiles and leave aside the interoperability of users contacts data. Indeed, interoperability of users' profiles is specifically related to the use of tokens [1] and Uri's [90, 61]. Commonly, a service, generally provided by an Identity Provider (IdP), stores and delivers user identifi-

[1] http://tools.ietf.org/html/rfc6749, last access May 2014

Table 2.1: Analysis of academic proposals related to access control in WBSNs (I)

Proposals		Analysed features								
		Elements in access control policies	Policy definition	Details on policy evaluation and enforcement mechanisms	Interoperable elements	Interoperable data	Sticky elements	Protection techniques	Protection elements	Trusted entity
[1]	M. P. Machulak et al. (2010)	-	XACML	Resource is granted if having the appropriate token	OAuth token	Policies, resources	-	-	-	Device (storage)
[24]	J. Li et al. (2009)	Role	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[27]	W.L. Fong (2011)	Relationship type	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[28]	B. Carminati et al. (2006)	Relationship type, trust, γ	N3-RDF	Resource granted if satisfying policies according to policy elements						
[29]	A. Masoumzadeh et al. (2010)	User attributes, data attributes	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[30]	J. Park et al. (2011)	User attributes, data attributes, conditions obligations	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[31]	P. Kumari et al. (2011)	Trust, γ	-	Resource granted if satisfying policies according to policy elements	-	-	Policies	-	-	Device(server)
[32]	H. Hu et al. (2012)	Relationship type	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[35]	L.M. Aiello et al. (2010)	Role	-	Resource is granted if having the appropriate token and verifying the regular expression attached to the token	Certificate	Resources	-	-	Key pair(private/ public) per user	-
[36]	A. Shakimov et al. (2011)	Role, location	-	Resource is granted if having the appropriate token and satisfying policies according to policy elements	Group descriptors	Resources	-	-	Key pair(private/ public) per user	Group of users +Device (provided by third parties)
[37]	M. Backes et al. (2011)	Role	Specific API	Resource is granted if having the appropriate token and verifying the token against a special list	Signed pseudonym	Resources	-	-	Not implicitly specified	X
[39]	K. Graffi et al. (2011)	-	-	Cryptographic algorithm	-	-	-	Cryptography	Key pair(private/ public) per user and key per object	-
[40]	M. M. Lucas et al. (2008)	-	-	Cryptographic algorithm with proxy cryptography to handle "one-to-many" requests	-	-	-	Cryptography	Key pair(private/ public) per user, other group pair and proxy keys per user per group	

*Data partially encrypted

Table 2.2: Analysis of academic proposals related to access control in WBSNs (II)

Proposals	Elements in access control policies	Analysed features: Policy definition	Details on policy evaluation and enforcement mechanisms	Interoperable elements	Interoperable data	Sticky elements	Protection techniques	Protection elements	Trusted entity
[54] A. Tapiador et al. (2012)	Role	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[55] C.W.D Munckhof (2011)	Data attributes	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[56] W. Villegas et al. (2008)	Trust, γ	-	Resource granted if satisfying policies according to policy elements used an ACL	-	-	-	-	-	-
[57] B. Ali et al. (2007)	Trust	-	Resource is granted if having the appropriate token	Keys	Resources	-	-	Key portions	Device (storage)
[58] B. Carminati et al. (2009)	Relationship tye, trust, γ	SWRL	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[60] L. M. Aiello et al. (2012)	-	-	Certificate validation	Certificates	Resources		Cryptography	Key per user	Group of users
[61] M. Ackermann et al. (2009)	-	-	Message interchange protocol	Similar to OpenId URI	Identity data	-	Cryptography*	Key pair(private/ public) per user	Group of users
[62] S. Guha et al. (2008)	-	-	Cryptographic algorithm based on dictionaries	-	-	-	Cryptography	Depending on web services	Group of users +Device (elements provided by third parties)
[63] W. Luo et al. (2009)	Role	-	Cryptographic algorithm	-	-	-	Cryptography	A master, an index and an access key per user and a master and index key per requester	Group of users + Device(server)
[64] T. Besenyei et al. (2011)	-	-	Cryptographic algorithm	-	-	-	Cryptography	-	-
[65] N. Kourtellis et al. (2010)	Trust, role, γ	-	Resource granted if satisfying policies according to policy elements and cryptographic public key algorithm	-	-	-	Cryptography	Key pair(private/ public) per object or a key per object	Group of users
[66] Y. Zhu et al. (2010)	-	-	Cryptographic algorithm based on data blocks	-	-	-	Cryptography	Key pair(private/ public) per user and other group pair	Group of users

*Data partially encrypted

Table 2.3: Analysis of academic proposals related to access control in WBSNs (III)

Proposals		Analysed features								Trusted entity
		Elements in access control policies	Policy definition	Details on policy evaluation and enforcement mechanisms	Interoperable elements	Interoperable data	Sticky elements	Protection techniques	Protection elements	
[67]	L. A. Cutillo et al. (2009)	Trust	-	Cryptographic public key algorithm and certificates	-	-	-	Cryptography	Key pair(private/ public) per user, attribute	Group of users
[68]	K. B. Frikken et al. (2009)	-	-	Cryptographic algorithm based on satisfying indirect paths	-	-	-	Cryptography	Key pair(private/ public) per user and as many keys as the maximum path length	Device(server)
[70]	R. Baden et al. (2009)	Role, γ	Using attributes and rights	Cryptographic algorithm based on CP-ABE	-	-	ABE keys	Cryptography	Key pair(private/ public) per user and a key per group	Device(Key storage)
[71]	S. Jahid et al. (2011)	Role	Using attributes	Cryptographic algorithm based on CP-ABE	-	-	ABE keys	Cryptography	Key pair(private/ public) per user and proxy keys depending on attributes and users	Proxy
[72]	S. Jahid et al. (2012)	Role	Using attributes	Cryptographic algorithm based on CP-ABE	-	-	ABE keys	Cryptography	Key pair(private/ public) per user depending on attributes and users and a signature key	-
[73]	S. Braghin et a. (2011)	Data attributes, γ	-	Cryptographic algorithm based on ABE with revocation	-	-	ABE keys	Cryptography	A master key and a key pair per established relationship	Group of users
[75]	R. Schlegel et al. (2012)	-	-	Cryptographic algorithm based on IBE	-	-	-	Cryptography	Key pair(private/ public) per user	Device(server)
[83]	Y. Cheng et al. (2012)	Relationship type, γ	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[85]	A. Besmer et al. (2009)	Role, user attributes		Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[86]	A. Tootoonchian et al. (2009)	Role	-	Resource is granted if having the appropriate token and verifying the the token against special list	Attestations	Resources	-	-	Key pair(private/ public) per user	
[87]	S. Buchegger et al. (2009)	Role	-	Cryptographic algorithm	-	-	-	Cryptography	Not explicitly specified	-
[88]	H. Zhang et al. (2012)	Role, user attributes, time period	XACML	Resource is granted if having the appropriate token	OAuth token	Resources	-	-	Key pair(private/ public) per user	Group of users

*Data partially encrypted

Table 2.4: Analysis of academic proposals related to access control in WBSNs (and IV)

		Analysed features								
	Proposals	Elements in access control policies	Policy definition	Details on policy evaluation and enforcement mechanisms	Interoperable elements	Interoperable data	Sticky elements	Protection techniques	Protection elements	Trusted entity
[89]	A. Ahmad et al.	Role	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[90]	S. Kruk et al. (2006)	Trust, γ	-	Resource granted if satisfying policies according to policy elements used an ACL	Password	Identity data	-	-	-	Group of users + Device (storage)
[91]	H. Wang et al. (2010)	Relationship type, trust, γ, obligations		Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[92]	S-W. Seong et al. (2010)	Data attributes		Resource is granted if having the appropriate token	Tickets	Resources	-	-	Key pair(private/ public) per user	Device (storage)
[93]	J. Park et al. (2000)	User attributes, data attributes, conditions obligations	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[94]	T. Abdessalem et al. (2011)	Relationship type, user attributes, γ	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[95]	I. B. Dhia (2012)	User attributes, relationship type, γ	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[96]	I. B. Dhia et al. (2012)	User attributes, relationship type, γ	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[97]	W.L. Fong et al. (2009)	Relationship type, γ	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[98]	G. Bruns et al. (2012)	Relationship type	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[99]	M. Alizadeh et al. (2012)	Relationship type	Answer Set Programming (ASP)	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
[100]	M. Conti et al. (2011)	-	-	Resource is granted if having the appropriate token	XML file	Resources	-	-	-	-
[101]	A. C. Squicciarini et al. (2009)	Role, user, attributes	XML	Resource granted if satisfying policies according to policy elements	-	Policies	Policies	-	-	Device (storage)

*Data partially encrypted

cations. For instance, in [61], OpenID[2], a decentralized identification standard is used by users to identify themselves through an URL. Similarly, [90] uses a specific service to manage identity based on the FOAF ontology. Note that OpenID and FOAF are described in Chapter 3.

As identity data, *interoperability* of access control policies has not been studied so far. The management of interoperable policies is pointed out in [101] but, surprisingly, policies have to be stored in every WBSN where users are enrolled in. On the contrary, in [1] policies are established in an independent entity in charge of enforcing access control and delivering tokens. Indeed, [1] applies User-Managed Access (UMA) protocol (described in Chapter 3). UMA facilitates the management of interoperable resources and access control policies simultaneously.

Only 6 proposals address the *sticky policy* requirement. Specifically, the majority of them make use of ABE in regard to an established group of attributes. Recalling that ABE cryptography involves specifying policies in keys or ciphertexts, it naturally helps to control access to data wherever it is used [70, 71, 72, 73]. Nonetheless, [101, 31] propose the development of a client-side mechanism, e.g. a plug-in installed in the browser, to enforce access control along the whole usage process and not only at data delivery. This mechanism verifies established access control policies either periodically or per user action, thus increasing the client computation costs in relation to access control management.

A total of 17 recent proposals make use of cryptography before storing data in the server. Therefore, they are in the position of fulfilling the *protection against honest-but-curious servers* requirement [62, 40, 67, 87, 63, 65, 66, 68, 70, 71, 72, 64, 61, 60, 39, 73, 75]. Detailed in Section 2.3, a total of five categories are distinguished concerning cryptography in access control for WBSNs. The overall procedure consists of using public-private key pairs in multiple and assorted algorithms, being key management one of the main problems at stake.

One last point of this analysis addresses the trust put by the system in sev-

[2]http://openid.net/get-an-openid/, last access May 2014

eral entities. In what concerns the *protection against honest-but-curious servers* requirement, the lack of cryptographic techniques requires trusting storage entities. However, there are some exceptions and several cryptographic proposals also trust particular devices but it is not due to confidentiality needs but to preserve data from unauthorized deletions [87] or from unfairly key managements [68, 75].

From this study several results are drawn. Firstly, the lack of use of assorted access control policy elements is identified. Only 5 approaches manage trust and a similar amount of them work with indirect relationships. WBSNs deal with an assorted set of characteristics such as users hobbies, users' age, the date of establishment relationships, etc., that are prone to be used in the access control enforcement process to achieve more fine-grained management. In second place, a relevant set of works are candidates for reaching interoperability but only regarding resources. Indeed, interoperability of identity data like users' contacts data is not even mentioned. Thus, proposals should work in the development of systems focused on interoperable and reusable identity data, resources and access control policies. Furthermore, it should be noticed that cryptographic approaches must relieve the burden of managing keys. As a final remark, sticky policies pose a tough challenge to address. Studied proposals focus on ABE cryptography but, as an alternative, a particular infrastructure could be deployed at the client-side to guarantee continuous policy verifications.

2.4.1.2 WBSNs in use

Many WBSNs are currently used by thousands of people. For the sake of simplicity, 9 of the most representative WBSNs in use have been studied according to features described above. The overall results are summarized in Table 2.5 which points out the simplicity of access control management procedures in contrast to academic proposals.

With respect to the *fine-grained* requirement, there is a common pattern in

Table 2.5: Analysis of access control features in WBSNs in use

	Analysed features								
WBSNs	**Elements in access control policies**	**Policy definition**	**Details on policy evaluation and enforcement mechanisms**	**Interoperable elements**	**Interoperable data**	**Sticky elements**	**Protection techniques**	**Protection elements**	**Trusted entity**
LinkedIn (2003)[1]	Role, *gamma* data privacy	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
Hi5 (2003)[2]	Role, data privacy	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
Facebook (2004)[3]	Role, location, data privacy	XACML	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
Orkut (2004)[4]	Role, email	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
Badoo (2006)[5]	Data privacy	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
Twitter (2006)[6]	Data privacy	-	Resource granted if satisfying policies according to policy elements	-	-	-	-	-	-
Picasa (2002)[7]	Data privacy, *gamma*	-	Resource granted if having the appropriate token	URL	Resources	-	-	-	-
MySpace (2003)[8]	Role, *gamma*, age, data privacy	-	Resource granted if satisfying policies according to policy elements or having the appropriate token	URL	Resources	-	-	-	-
Flickr (2004)[9]	Role, *gamma*, data privacy	XACML	Resource granted if satisfying policies according to policy elements or having the appropriate token	URL	Resources	-	-	-	-

[1] http://www.linkedin.com/ , last access May 2014
[2] http://www.hi5.com/ , last access May 2014
[3] http://www.facebook.com/ , last access May 2014
[4] http://www.orkut.com/PreSignup , last access May 2014
[5] http://badoo.com/ , last access May 2014
[6] https://twitter.com/ , last access May 2014
[7] https://picasaweb.google.com/home , last access May 2014
[8] http://www.myspace.com/ , last access May 2014
[9] http://www.flickr.com/ , last access May 2014

access control policies that consists of using roles and data privacy. Note that data privacy corresponds to mark data as public, private or restricted to a set of users. Besides, all analysed WBSNs manage bidirectional relationships, except for Twitter and MySpace. In the former, administrators request users to be followers of them but not necessarily in the other way round. In the latter, MySpace, administrators make (out connections) or receive (in connections) friendship requests and they may accept them, distinguishing between in, out and in-out connections. By contrast, apart from relationship's direction, other elements such as relationships creation time, size of data or users nationality, are neglected.

Three WBSN, namely, Picasa, MySpace and Flickr, manage tokens and thus, are candidates for addressing the *interoperability* requirement. Picasa, focused on photos management, applies tokens which take the form of URLs. Likewise, Flickr,

that also focuses on photo management, uses URLs to provide access to photos. This WBSN is specially relevant for public photos management. Everybody is able to access some selected photos through an URL. Moreover, Flickr allows the management of rights in regard to notes, commentaries and photos. Similarly, MySpace, a WBSN to share data such as photos, videos or music, uses URLs as tokens as well. However, MySpace only uses tokens for photos, applying simpler techniques for other types of data like wall messages. Therefore, tokens seem to be specially valuable for photo management.

Furthermore, regarding trusted entities, WBSNs do not detail whether some entity or object is trusted or not. Then, as there is no evidence of the fact that WBSNs address *protection against honest-but-curious servers*, it is concluded that trust should be put into servers, as well as, in users with whom relationships are established.

In conclusion, WBSNs in use provide certain *fine-grained* management. Moreover, as in academic proposals, roles are the main elements of access control policies. On the other hand, some WBSNs in use apply tokens and can be candidates for providing resource *interoperability*. Nonetheless, they do not provide real interoperable functionalities such as the visualization of photos of one WBSN in another. Finally, it is remarkable that nothing is mentioned in regard to the *sticky-policies* requirement.

2.5 Expressive power of ACMs for WBSNs

The management of three of the identified requirements (*interoperability*, *protection against honest-but-curious servers* and *sticky-policies*) has been studied in depth. However, the *fine-grained* requirement needs further analysis. As stated in Section 2.4, an ACM, to be expressive in the WBSN field, has to facilitate the management of the following six features: (F1) *distance*, (F2) *common-contacts*, (F3) *clique*, (F4) *multi-path*, (F5) *direction* and (F6) *flexible elements in access control policies*.

Expressive power means the ability of ACMs to support a wide variety of policies. More specifically, the expressive power of an ACM is a measure of the range of supported policies. Given a pair of models (A and B), if all policies that can be represented in B can also be represented in A, then A is at least as expressive as B [102, 79]. Note that though each considered model is attached to a particular policy language and the expressive power of the model is analysed using that policy language, in this work expressive power is referred to the ACMs.

Techniques to compare expressive power of ACMs have received some attention, though not particularly focused on ACMs for WBSNs. For instance, Ganta *et al.* present a formalization to compare expressive power focused on Typed Access Matrix Model (TAM), Augmented TAM and their variations [103]. Similarly, Bertino *et al.* present a framework for reasoning about ACMs [102]. In this case, the framework focuses on mapping rules (composed of objects, subjects, privileges and authorizations) of a couple of models to a common language based on mathematical logic, and comparing results to determine the model that is at least as expressive as the other one. Afterwards, Tripunitara *et al.* propose a theory for comparing models based not only on the state of the model, expressed by a particular access control policy, but on the state-transition [79]. Their proposal expresses an ACM as a set of states, policies and state-transitions rules to define how to pass from one state to another. With the same purpose but from a different perspective, several approaches apply simulation techniques to perform the comparison [104, 105].

On the other hand, some contributions analyse several features of access control in WBSNs, though not being particularly related to expressive power. Carminati and Ferrari present a survey of access control in WBSNs that studies features such as the kind of relationships existing in current WBSNs and the applied management procedures [17]. From a more specific point of view, A. Lazouski *et al.* survey literature related to $UCON_{ABC}$ usage model, studying among other aspects managed elements in access control policies [20]. Moreover, Cheng *et al.* point out the

necessity of achieving expressive fine-grained policies and study characteristics of different ACMs for WBSNs like the management of multiple relationships types [83].

In this work, the comparison of expressive power of ACMs for WBSNs has been done by analysing whether the policy languages linked to each ACM are able to express a set of predefined policies P1-P7. Next, Section 2.5.1 describes the motivation and details the applied methodology. Section 2.5.2 presents the proposed set of policies P1-P7, applying inductive reasoning to assure that they are representative of a general case of some of the considered features (see Appendix C). Section 2.5.3 summarizes the analysis of 24 ACMs (the details of the analysis can be found in Appendix C).

2.5.1 Motivation and methodology

Bertino *et al.* proposal inspires the comparison performed herein. They propose a framework to homogeneously represent ACMs to be later compared [102]. In particular, models are represented under a common logic language and their expressive power is compared by analysing their *structural* and *access equivalence*. The former refers to verifying if models are built from the same set of structural components and the latter to checking whether models instances enforce the same set of accesses.

Nonetheless, Bertino *et al.* framework is not appropriate for WBSNs because it is specially focused on DAC, MAC and RBAC, thereby limited for the large quantity of WBSN access control management elements. This framework manages access control policies (called authorizations) composed of objects, subjects and privileges, which are a reduced set of elements for the WBSN field. For instance, relationships are essential WBSN elements, as well as subjects, objects or relationships attributes, and they are hard to manage within Bertino *et al.* framework in its current status. Then, instead of comparing *structural* and *access equivalence*, to analyse and compare the expressive power of WBSN ACMs, where a varied set of

access control elements is at stake and an assorted set of policies can be considered, the *semantic equivalence* of access control policies is compared herein.

Given the definition of a policy to be constructed which is composed of several features, *semantic equivalence* is defined as the identification of the fact that access control policies constructed by using the policy language of every ACM to analyse, can express all features involved in the proposed policy. For instance, the policy "*grant access to friends of a friend*" can be expressed in multiple policy languages but the semantic meaning should be analogous in all of them. Thus, ACMs that own policy languages which allow the specification of a given access control policy (pointed out as expressive) would be considered ACMs with expressive power regarding features attached to the established policy. Indeed, thanks to the proposed semantic equivalence technique, the expressive power of ACMs is assessed without the need of translating them into a common, unified representation.

The comparison methodology proposed herein consists of several tasks. Initially, concerning the *fine-grained* requirement, seven access control policies to cover all identified features (*distance*, *common-contacts*, *clique*, *multi-path*, *direction* and *flexible elements in access control policies*) are proposed in Section 2.5.2. Applying inductive reasoning, it is asserted that the model that expresses a policy linked to a particular feature can express any policy associated with this feature (see Appendix C.1). However, inductive reasoning is not applied to *direction* (F5) and it would be extremely tedious in respect to *fine-grained* (F6) due to several reasons. Regarding F5, this feature exclusively requires the creation of directional and bidirectional relationships and consequently, generalization does not have to be applied. By contrast, according to F6, the amount of attributes that can be managed is extremely varied and their generalization is unattainable. Subsequently, 24 approaches are analysed to establish whether the definition of the proposed access control policies using the policy language provided in each work is possible (see Appendix C.2). Finally, a summary and a discussion are presented according to the *semantic equiv-*

alence analysis of each studied proposal. The mentioned set of proposals focus on ACMs for WBSNs as well as mechanisms that, without proposing a specific model but being based on a particular one, contain a policy language. Besides, based on Section 2.2, these approaches are classified under RBAC, RelBAC, ABAC, TBAC and OBAC.

2.5.2 Specification of access control policies used to evaluate expressive power

According to WBSNs features, namely, (F1) *distance*, (F2) *common-contacts*, (F3) *clique*, (F4) *multi-path*, (F5) *direction* and (F6) *flexible elements in access control policies*, it is considered that, in a WBSN, an administrator establishes the following set of access control policies to grant access to a particular data to a requester. Note that some of the proposed access control policies involve the management of more than a single feature. Additionally, for the sake of simplicity, all models are assumed to exclusively provide the right to access an object leaving aside other rights, such as writing or deleting. Notice that rights can be *grant* or *deny* but the latter one is not managed because rejections can be expressed applying negatives. For instance, "*deny access to contacts considered friends*" is analogous to *"grant access to contacts not considered friends*".

The proposed policies are the following ones:

P1 Access is granted to users who are friends of neighbours of his/ her relatives if the relationship between his/ her relatives and his/ her relatives' neighbours was established before year 2,000. (F1 and F6)

P2 Access is granted to users who have three friends in common with the administrator of the requested object. (F2)

P3 Access is granted to users who belong to a clique in which two users and the administrator of the requested object are involved, having all of them a friendship relationship. (F3)

P4 Access is granted to users who are connected to the administrator by two different paths composed of unidirectional relationships oriented from the requester to the administrator. Moreover, relationships involved in all paths have to be highly trusted. (F4 and F6)

P5 Access is granted to users who are friends of the administrator of the requested object, also having a bidirectional relationship with him/ her. (F5)

P6 Access is granted to users who are friends of the administrator of the requested object. (F5)

P7 Access is granted to users if they are females under 30 years old or if they are females under 40 who have studied computer science or if they are females who have studied computer science and physics. (F6)

2.5.3 Analysis of the expressive power of ACMs for WBSNs

Once concluded the evaluation of the expressive power of a set of 24 ACMs for WBSNs, detailed in Appendix C.2, in this Section a summary of the conclusions is presented. The most appropriate model corresponds to the one that allows the specification (based on the proposed model-dependant definitions) of as many policies as possible, facilitating the expression of a wide set of user preferences. Table 2.6 presents a summary of the analysis. Each policy is marked as *completely* ($\surd$) or *partially* ($\mathcal{P}$) defined according to the achieved expressive power. Moreover, note that policies that involve more than a single feature, P1 and P4 in particular, require $\surd$ or $\mathcal{P}$ per each involved feature. No marks are used for undefined features.

The most surprising issue is that, even being models specially focused on WBSNs, none of them allows the expression of all identified features (Section 2.4), thus limiting the possibilities of users to manage their personal preferences. Furthermore, concerning the ACMs classification (RBAC, RelBAC, ABAC, TBAC and OBAC), models based on roles seem to be the least expressive while those that

Table 2.6: Expressive power comparison of ACMs

WBSN Models	P1 (*Distance*) – (*Flexible elem. in a.c.p.*)	P2 (*Common-contacts*)	P3 (*Clique*)	P4 (*Multi-path*) – (*Flexible elem. in a.c.p.*)	P5 (*Direction:Bidi.*)	P6 (*Direction:Unidi.*)	P7 (*Flexible elem. in a.c.p.*)
RBAC							
[24] J. Li et al.					√	√	
[54] A. Tapiador et al.					√	√	
[89] A. Ahmad et al.						√	
TBAC							
[57] B. Ali et al.				– P	√		
[28] B. Carminati et al.	√ –			– P		√	
[94] T. Abdessalem et al.	√ –			– P	P	√	P
[56] W. Villegas et al.	P –			– √		√	
[91] H. Wang et al.	P –			– √		√	
RelBAC							
[27] P. Fong et al.	√ – P	√	√	√ –	√	√	
[97] P. Fong et al.	P	√	√	P –	√		
[83] Y. Cheng et al.	√ – P				√		
[32] H. Hu et al.	P				√		
[95] I.B. Dhia	√ – P			– √	√	√	P
[96] I.B. Dhia	√ – P			– √	√	√	P
[98] G. Bruns et al.	√ – P	√		P –	√	√	
ABAC							
[93] J. Park et al.	– √	√	√		√	√	√
[30] J. Park et al.	– √	√	√		√	√	√
[55] CVD. Munckhof						√	√
[71] S. Jahid et al.	– P					√	
[70] R. Baden et al.	– P					√	
[73] S. Braghin et al.	√ – P					√	
OBAC							
[29] A. Masoumzadeh et al.	√ – P	√	√	P –	√	√	P
[58] B. Carminati et al.	√ –					√	
[99] M. Alizadeh et al.						√	

Policies

√ : A feature is completely expressed

P : A feature is not completely expressed

manage relationships and attributes are the most expressive ones.

According to indirect relationships (F1 and P1), the great majority of models deal with them. More specifically, 10 out of 24 achieve the successful definition of indirect relationships with an unlimited distance between users [29, 28, 94, 27, 83, 58, 98, 95, 96, 83]. By contrast, [97] and [32] only define two hops relationships.

Concerning common contacts (F2 and P2), a total of 6 models out of 24 manage this feature [93, 30, 29, 27, 97, 98]. Nonetheless, even attaining a successful definition of P2, it would be desirable to work in describing access control enforcement procedures similarly to [98].

A challenging issue is the specification of cliques (F3 and P3) because their management involves a great deal of complexity. Surprisingly, this feature is managed by the same models that deal with F2, except for [98]. As highlighted in [80], users involved in a clique have to be discovered from unreachable users and the difficulty of its management is not even mentioned in Fong *et al.*'s proposals. Besides, no

guidelines regarding access control enforcement are provided.

On the other hand, multi-path (F4 and P4) is other appealing feature that 4 models out of 24 manage [29, 27, 97, 98]. Nevertheless, three of them ([29, 97, 98]) do not fully deal with F4 as considered in P4. They cannot define policies which include multiple and different paths. Indeed, satisfying F4 is a challenging matter in the WBSN context [82, 17], although it can be simplified to groups management. For instance, the existence of the relationship "relative" and the relationship "friend" can be compared with the creation and management of a group of friends and a group of relatives.

Relationships direction is another studied feature (F5, P5 and P6). Concerning bidirectional relationships (F5 and P5), they are interestingly managed by 13 out of 24 proposals. Assorted techniques are applied to deal with this type of relationships: specific attributes are created [93, 106], pairs of directional relationships are established [29, 94, 98] or, as in the majority of current WBSNs, relationships are considered inherently bidirectional [24, 97, 32]. However, notice that [94] unsuccessfully defines P5 since the created policy is satisfied by unidirectional and bidirectional relationships. On the other hand, in what concerns unidirectional relationships (F5 and P6), 20 out of 24 approaches manage them. Those that are based on cryptography require exchanging decryption keys and thus, unidirectional relationships are implicitly established [95, 96, 73]. By contrast, other approaches like the one proposed by M. Alizadeh *et al.* [99] highlight the management of directed labelled relationships.

In regard to flexible elements in access control policies (F6, P1, P4 and P7), the models proposed by J. Park *et al.* [93, 30] are the most expressive ones, attaining the proper specification of P1 and P7. According to P1, fine-grained management is considered in [29, 28, 27, 83]. In particular, these proposals particularly focus on relationships roles. On the other hand, concerning P4 and meeting expectations, TBAC models express the proposed trust relationship. Finally, only J. Park *et al.*

models [93, 30] successfully achieve the specification of P7. In general, except for J. Park *et al.* models [93, 30], ACMs do not include disjunctives management and then, the creation of as many access control policies as sentences connected by disjunctives is missing. Furthermore, the management of multi-valued attributes is, again, only managed by [93, 30]. A. Masoumzadeh *et al.* [29] and T. Abdessalem *et al.* [94] models could be easily modified to manage this kind of attributes but currently, they do not deal with them. As a final remark, despite the unreachable generalization of F6 (pointed out in C.1), much more improvements must be performed to express varied preferences using, simultaneously, disjunctives and conjunctives together with different attributes.

In the light of this analysis, the models proposed by A. Masoumzadeh *et al.* [29], P. Fong *et al.* [27] and J. Park *et al.* [93, 30] are the most expressive for social networking applications. Their level of *semantic equivalence* is significant as they allow the definition of a lot of features. The first pair of proposals focus on relationships while the latter are based on attributes management. Thus, it is asserted that the development of expressive ACMs for WBSNs has to go towards the management of relationships, as well as the management of attributes of users, objects and relationships.

Moreover, given that J. Park *et al.*'s models [93, 30] are based on $UCON_{ABC}$ and it is a mature model which lays the bases of the first contribution of this thesis, its description is presented in the following section.

2.5.3.1 $UCON_{ABC}$ model

The $UCON_{ABC}$ model considers eight components: *subjects (S)*, that are entities that exercise rights on objects; *objects (O)*, that are entities which subjects hold rights on; *subject attributes (ATT(S))* and *object attributes (ATT(O))* that refer to features associated with subjects and objects, respectively; *rights (R)*, which are recognized as privileges exercised on objects such as read or write; *Authoriza-*

tions (A), that correspond to predicates on subject and object attributes that are evaluated in order to decide whether the requested right on a specific object made by a certain subject should be allowed or denied; *oBligations (B)*, that represent predicates that must be satisfied before, during or after the right is granted; and *Conditions (C)*, that correspond to environmental or system factors which are taken into account during the access decision process.

Additionally, $UCON_{ABC}$ introduces a pair of decision properties that are applied in respect to A, B and C. First, *continuity* refers to the enforcement of the usage decision along the whole usage period (pre/ ongoing). Second, *mutability* refers to changes produced in attributes along the usage process (pre/ ongoing/ post). Based on both features, *continuity* and *mutability*, the usage has to be revoked when policies become unsatisfied. Pre/ ongoing authorizations ($preA$/ onA), as well as pre/ ongoing conditions and obligations ($preB$/ onB and $preC$/ onC respectively) are applied in $UCON_{ABC}$ to manage usage control. In this regard, a Usage Decision Facility (UDF) and a Usage Enforcement Facility (UEF) which are always active [107] are applied in the usage control process. UDF identifies changes in attributes and UEF enforces access control accordingly.

In the original $UCON_{ABC}$ model, it is assumed that an access control policy is defined by the system's administrator and this policy is applied to all users in the system. A recent work by Salim *et al.* propose an administrative model, orthogonal to the $UCON_{ABC}$ model, where the attributes and rights of subjects and objects are established through assertions made by authorized subjects [106].

2.6 Access control administration in collaborative environments.

This Section introduces the essential points of access control administration, which are the context of this thesis. First, administrative tasks are introduced (Section 2.6.1). Second, types of administration, as well as positive and negative points of

each of them according to WBSNs are described (Section 2.6.2). Finally, an analysis of administrative management features is presented (Section 2.6.3).

2.6.1 The power of administration

Access control administration ensures that users do not make unauthorized access requests [108]. From traditional systems where a single administrator manages the permissions of all users, as well as from systems where users manage permissions of the data they create, e.g. UNIX operating system, the development of collaborative systems leads to reconsider administrative tasks.

Coming back to the 90's, given the maturity of the Role Based Access Control Model (RBAC) proposed by R. Shandu *et al.* [109], its attached administrative model can be used as a precedent in the identification of administrative tasks [110]. In a nutshell, in RBAC, administrative permissions (analogous to rights) are exclusively applied to administrative roles and other permissions are applied to any other kind of roles. Then, administrative tasks are based on the assignment of users to roles; the assignment of permissions to roles; and the assignment of roles to roles. Administrative tasks are summarized as follows:

- Who is the entity in charge of creating, updating and deleting access control preferences.
- Who is the entity in charge of associating preferences with data.
- How preferences are associated with data and data with data owners.

Furthermore, administrative issues also involve administrative rights management. Two types of rights are distinguished, namely, use and administrative rights. Use rights consist of operations performed with objects, e.g read right, and administrative rights correspond to operations performed over the right of objects, e.g. the right to give read right. The management of both types of rights is essential and delegation and revocation are remarkable operations in this regard. Delegation

focuses on granting a right to a user, while revocation undoes the effects of delegation. In particular, *weak* and *strong revocation* are differentiated. The former refers to simply remove granted permissions and the latter refers to recursively revoke permissions from those to whom the grantee granted the permissions. Based on these rights and operations, the following administrative tasks are added to the previous ones:

- Who is the entity in charge of managing revocation.
- Who is the entity in charge of managing delegation.
- How weak and strong revocation are managed based on use rights and administrative rights.
- How delegation is managed based on use rights and administrative rights.

In the social networking field administration focuses on managing uploaded resources, specified identity data and established access control policies (recall Section 2.1). Thus, WBSN administrative tasks are equivalent to the ones above mentioned but considering that resources, identity data and policies are the elements at stake.

2.6.2 Types of administration

From the traditional point of view, types of administration can be classified in two categories *centralized* and *decentralized* [111]. In the following, administration types together with pros and cons of each of them, are described:

- *Centralized*: a single entity decides who can get into the system. Administrators configure the system deciding which users can get access to which data.

  On the positive side, this is the simplest administration type. It helps to reduce management costs because except for the administrator, nobody is

involved in the administration process. However, it is not specially appropriate for WBSNs because in this area users desire to individually manage their data.

- *Decentralized*: multiple entities decide who can get into the systems. Access requests are administered by users instead of by a central entity. Besides, in some cases data management may involve multiple users, e.g. in a photo several users can be depicted, and apart from the owner, other users are pointed out as administrators too. These users, in the role of administrators, are called co-owners.

  Positive points of decentralized administration focus on the possibility of collaboration in social environments. Decentralization facilitates that each WBSN user manages his data. However, on the negative side, the complexity of access control management increases. Even more, in case co-ownership management, preferences of owners and co-owners have to be jointly evaluated and all possible conflicts have to be identified and solved.

2.6.3 Analysis of access control administration in collaborative environments

Based on identified administration types and tasks, this Section analyses 21 proposals in the literature that address administrative issues in collaborative environments. Note that this study is not exclusively focused on WBSNs, but extended to collaborative environments due to a pair of reasons. On the one hand, WBSNs manage data which may be related to multiple users and then, they can be pointed out as collaborative systems. On the other hand, a small amount of proposals focus on administrative issues in the specific context of WBSNs.

In general, 6 contributions fall in the WBSN category [58, 112, 82, 113, 114, 115], 3 proposals in document sharing [116, 117, 118], one proposal is based on grid environments [119] and the rest of them focus on other general collaborative systems

Table 2.7: Administrative features analysis

	Proposals	Administration	Delegation	Revocation
[59]	B. Carminati et al. (2011)	D		
[82]	A.C. Squicciarini et al. (2009)	D		√*
[88]	H. Zhang et al. (2012)	C		
[112]	M.R. Thompson et al. (2003)	D	√	√
[113]	A.C. Squicciarini et al. (2010)	D		√*
[114]	A. Ahmad et al. (2012)	D	√	√
[115]	Y. Jung et al. (2013)	D		√
[116]	Y. Ren et al. (2011)	D		
[117]	M. Prilla et al. (2006)	D		√
[118]	A. Imine et al. (2009)	D		√
[119]	M. Lorch et al. (2003)	D	√	√
[120]	H.F. Wedde et al. (2003)	D		
[121]	R. S. Shandu et al. (2010)	D	√	√
[122]	R. S. Shandu et al. (2011)	D	√	√
[123]	W.K. Edwards (1996)	C		
[124]	K. Sikkel et al. (1997)	D	√	√
[125]	Z.Y. Zhang et al. (2011)	D	√	√
[126]	R.K. Thomas (1997)	C		
[127]	E. Cohen et al. (2002)	D	√*	
[128]	V. Gligor et al. (2002)	D		√*
[129]	J. Jin et al. (2006)	D	√	

∗: mentioned but not managed

[120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 88].

This analysis studied the administration type, namely, centralized (C) or decentralized (D) and how delegation and revocation are managed. Table 2.7 presents results of the analysis. Symbol * means that a particular feature has been mentioned but not managed.

In what concerns the administration type, 18 approaches deal with D administration and just 3 proposals focus on C administration. As expected, administration tends to be decentralized (recall Section 2.6.2).

Concerning centralized administration, in [123] a central administrator manages roles and policies. Furthermore, the need of dynamism is highlighted and the change of user roles, at runtime, is an essential matter to deal with. Similarly, [126] proposes team management. Teams are composed of users with the same role whose management is left to a general administrator. Likewise, in [88] groups are managed by a central authority in such a way that users are added to groups and rules, based on user attributes, time periods and resource usages, are applied to

groups.

The majority of approaches are based on decentralized administration, allowing users to individually manage their personal data. For instance, in [118], the administrators initiate the administration process by notifying updates to affected users who become involved in the administrative management process. By contrast, in [129, 112] users who want to become involved in a particular administrative process have to request it. Other proposals divide data, particularly documents, among users and they work over each owned piece of data [116]. A different solution are proposed by M.R. Thompson *et al.* [112] and A. Ahmad *et al* [114]. M.R. Thompson *et al's* work is based on certificates jointly signed by all users involved in the administrative process. However, A. Ahmad *et al* propose *transfer*, *multiplication* and *division* operations [114].

Delegation, associated with decentralized administration, is addressed in a total of 9 approaches. In collaborative environments several users have to cooperate to achieve a common goal. Then, delegating permissions breaks the power of a central administrative user by sharing administrative tasks among different parties. The most of approaches focus on permissions delegation [121, 122, 124, 119, 112, 88, 114], being the proposals of Z.Y. Zhang *et al.* and J. Jin [129] the only ones which propose role delegation [125] and E. Cohen *et al.*'s proposal which exclusively mentions the difficulty in managing delegation in organizational environments [127].

Related to revocation management, in multiple cases users may regret having granted a certain use or administrative right to a user. A total of 10 proposals provide mechanisms to deal with revocation and other 3 contributions mention the relevance of its management [128, 82, 113]. They focus on weak revocation in respect to rights [121, 122, 117, 119, 114] and group memberships [118] and on strong revocation regarding delegated rights [124, 125, 115] and certificates [112].

In sum, administration in collaborative environments tends to be decentralized. This is specially remarkable in WBSNs as they involve the management of lots

of users and data. Besides, most of analysed approaches propose revocation and delegation mechanisms, thereby highlighting the relevance of their management as part of the administration process.

2.7 Dealing with co-ownership in access control systems for collaborative environments.

WBSNs make use of data that can be usually associated with multiple users. Then, joint ownership, also referred to as co-ownership, is pointed out as a demanding necessity. In relation to this issue, this Section presents the concept of co-ownership in access control systems (Section 2.7.1) and the analysis of how a significant amount of proposals in the literature deal with co-ownership (Section 2.7.2).

2.7.1 Access control management and administration of co-ownership

Data owners, also known as administrators, upload data to WBSNs or write or use them within its personal WBSN profile. Nonetheless, these data may be related to multiples users who are referred to as co-owners (recall Section 2.6.2).

Due to the aforementioned necessity of joint ownership, access control should consider the identification of co-owners along with the management of owners and co-owners access control preferences [89]. However, user preferences may be contradictory, e.g. a photo owner may want to make a photo public and a co-owner may prefer to keep it private. In view of this situation, negotiation mechanisms to deal with conflicts of interests are needed. For instance, a negotiation technique to establish preferences according to the most voted preferences is a common solution [130].

Access control administration cannot be disregarded. Co-ownership management requires that multiple users administrate access control, thereby becoming administration more difficult. All co-owners and the owner of a particular data

Table 2.8: Co-ownership management analysis

	Proposals	Neg. Mechanisms	Management elem.
[32]	H. Hu et al. (2012)	√	Groups membership
[33]	A.C. Squicciarini et al. (2011)	√	Groups membership
[34]	K. Thomas et al. (2010)	√	General conditions
[41]	F. Zhu et al. (2008)		Roles
[48]	H. Hu et al. (2011)	√	Groups membership
[82]	A.C. Squicciarini et al. (2009)	√	Users depth
[83]	Y. Cheng et al. (2012)	√	User relationships
[112]	M.R. Thompson et al. (2003)		Roles
[113]	A.C. Squicciarini et al. (2011)	√	Users depth
[116]	Y. Ren et al. (2011)		Groups membership
[117]	M. Prilla et al. (2006)	√*	Roles
[120]	H.F. Wedde et al. (2003)	√	Roles
[128]	V. Gligor et al. (2002)	√	Users and object attributes
[130]	B. Carminati et al. (2011)	√	Users trust and depth
[131]	Q. Xiao et al. (2012)	√	Depth, groups membership
[132]	A. Besmer et al. (2010)	√*	Groups membership
[133]	R. Wishart et al. (2010)	√*	General conditions
[134]	Y. Sun et al. (2010)	√	Users trust
[135]	D. Lin et al. (2008)	√	General conditions

*: mentioned but not managed

should take part in the administration process. Particularly, administrative tasks include the specification of how co-owners and owners preferences are established, as well as, how they are evaluated. Likewise, revocation and delegation should be also addressed in this new scenario.

2.7.2 Analysis of co-ownership management and administration in WBSNs

To have a general overview about joint ownership, apart from studying co-ownership management in WBSN, this Section analyses co-ownership in other kind of collaborative systems. A total of 19 proposals are analysed, examining a pair of features, the application of negotiation techniques and the type of elements involved in access control policies. Note that revocation and delegation are not studied because their application does not differ from the one presented in Section 2.6.

From the whole set of proposals just 10 contributions fall in the WBSN category

[130, 83, 32, 48, 112, 82, 113, 33, 131, 34], 3 proposals fall in documents sharing [116, 117, 41] or the remaining set fall in general collaborative systems [120, 128, 132, 133, 134, 135]. Results of the analysis are depicted in Table 2.8. Note that symbol * means that a particular feature has been mentioned but not managed.

First, concerning negotiation techniques, it should be noticed that collaborative environments such as WBSNs may produce conflicts of interests between users caused by contradictory preferences. In order to cope with this matter a great variety of negotiation mechanisms are proposed. Indeed, 3 approaches mention the necessity of negotiation [117, 132, 133] and 13 approaches propose varied solutions. In general, the most common negotiation technique is based on voting schemes [120, 32, 128, 130, 134, 33]. Given a set of preferences, the number of votes that each of them receives is used to calculate which preferences apply. For instance, in [130] access is granted regarding the satisfaction of all, one or the majority of established owners and co-owners preferences.

Similarly, H. Hu *et al.* proposes a solution where owners and co-owners preferences are managed as sets [48]. In case a conflict of interest appears due to existence of a subset, a superset, a partial set or a disjoint set of user preferences, measures of the privacy risk and the sharing loss (factors affected by the loss of sharing a piece of data) help to determine the preferences to apply. Following similar bases, [83] presents three intuitive techniques focused on the disjunction, the conjunction or the priority order of operators applied to user preferences.

From other point of view, A.C. Squicciarini *et al.* work in the automatic detection of access control preferences calculating the similarity between tagged users [82, 113]. Thus, automatic tools are used as negotiation mechanisms. The main drawback of this work is that users do not specify their preferences and automatic tools may do not really satisfy users expectations.

Other approach, though quite restrictive, is developed by D. Lin *et al.*, where four negotiation possibilities are offered [135]: *permit-overrides*, if a user's prefer-

ences are satisfied, the access is granted; *deny-overrides*, if a user's preferences are not satisfied, the access is denied; *first-one-applicable*, the first suitable user's preferences are the ones applied; and *only-one-applicable*, a single user's preferences are applied.

Nonetheless, all mentioned negotiation techniques have a common point of failure. Owners and co-owners preferences may be contradictory and then, some users privacy may become compromised, thereby violated. Trying to reach a full consensus among owners and co-owners preferences, Q. Xiao *et al.* proposes CAPE, a mechanism based on managing personal opinions and *peer effects* [131]. Users adjust their access control preferences regarding decisions of other users until a consensus is reached. However, the impossibility of achieving a full consensus is discussed when a user does not change his preferences even having taken into account other users' decisions. Indeed, K. Thomas *et al.*'s approach is the only one that completely respects users privacy [34]. The solution consists of calculating the intersection of all user preferences to grant or deny access accordingly.

The second and last point of analysis is the type of elements involved in access control policies. A significant percentage of approaches (11 out of 33) focus on roles management [122, 117, 124, 120, 125, 41, 126, 123, 127, 129, 112]. Users are assigned to roles with a set of permissions and they manage access according to roles they belong to. Similarly, some proposals use groups. Being member of a group means having a certain amount of permissions over a set of resources [118, 136, 116, 32, 48, 132, 33, 88, 131]. For instance, in [118] groups are composed of an administrator and several users. Administrators specify access control preferences to establish the users who can read and write over a document. It should be noticed that groups are pre-established in some contributions. In H. Hu *et al.*'s proposal there are four pre-established groups [48, 32] (presented in Section 2.1): *disseminators*, *contributors*, *owners*, *accessors*. Likewise, in [132, 33] users are divided in two groups, owners and co-owners.

On the other hand, an appealing issue in the WBSN context is the management of trust and depth between users [59, 137, 130, 134, 82, 113]. Users put trust in their contacts with whom are connected at different depths (e.g. in Facebook the maximum depth is 2, friend-of-a-friend). Note that depth is analogous to feature *distance*. In addition, the use of users and objects attributes is applied in a couple of approaches [119, 128]. Finally, a total of 3 contributions leave open the set of applied access control management elements [121, 133, 135].

In the light of this analysis, it can be concluded that a significant set of approaches work with owners and co-owners but management procedures may violate users privacy. The voting scheme is the most common negotiation technique but contradictory preferences may compromise users privacy. Indeed, just K. Thomas *et al.* propose the intersection of all user preferences to completely preserve all users privacy. However, this solution is significantly restrictive because access is denied unless all users reach a full consensus.

Chapter 3

Interoperability and reusability management. Applications and technologies.

Several applications and technologies facilitate interoperability to give access to data managed in one place from another. In this regard, reusability is a significant feature to consider as, for instance, reusable profiles would let users manage their identity data as a global identity. Reusability could be a powerful complementary feature to interoperability because if a pair of elements are interoperable, they can be analogously used and then, reused. Nonetheless, despite the benefits that reusing, i.e. saving storage, and interoperability, i.e. simplicity of data management, can jointly provide, they have not been mentioned together.

Concerning the social networking field, only a pair of distributed WBSNs have paid attention to interoperability, namely, Diaspora and Friendica. Furthermore, there are well-known protocols, such as OAuth, applied to the development of distributed systems which are a key step towards interoperability. Moreover, apart from protocols, the definition of interoperable, as well as reusable file formats is a necessary requirement. Particularly, the description of people and their relationships are essential WBSN data which should be reusable and interoperable.

According to mentioned above, this chapter presents, in Section 3.1 the definition of interoperability. Section 3.2 describes reusability. Next, Section 3.3 describes

a couple of distributed WBSNs focused on interoperability. Section 3.4 introduces protocols to develop distributed systems. Finally, Section 3.5 presents a pair of file formats which may be applied for interoperable and reusable purposes.

3.1 Interoperability definition

In 1990 interoperability was defined as "*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*" [138]. More specifically, in the technological area, interoperability is defined as "*the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units.*"(ISO/IEC 2382-01). In this regard, given the amount of software and hardware daily developed, lots of standards have been proposed, e.g. send meeting requests and other calendar data to other internet users [139], being organizations like the International Organization of Standardization (ISO) in charge of this process.

Moreover, the technological area involves persistent enhancements and developments of systems and applications, thus requiring the continuous search of interoperability. This is the case of WBSNs, there are a great amount of them and users are forced to be enrolled in all WBSNs that they want to enjoy. In particular, the interoperability problem in WBSNs is known as "the walled garden" [140]. WBSNs are depicted as walled places where it is extremely hard to climb over a wall to reach another (see Figure 3.1). Pointed out in Chapter 2, interoperability in WBSNs should be related to elements managed in them, namely, identity data, resources and access control policies. Interoperable identity data would allow users to specify their profiles and contacts in a single WBSN, being accessible in all of them in which users are enrolled. Similarly, resources interoperability would be extremely comfortable from the users point of view. It would allow the upload of resources in a WBSN remaining them available in other WBSNs. Besides, in terms

Figure 3.1: Walled garden problem in WBSNs

of privacy, access control policies interoperability is specially appealing. It would reduce the burden of specifying what is accessed by whom per resource and WBSN. Therefore, it would help to reduce users errors caused by repetitive actions.

3.2 Reusability definition

From the software engineering point of view, reusability is defined as "*the use of existing software components to construct new systems*" [141]. By contrast, in WBSNs, reusability can be described as the reuse of identity data, resources and access control policies in as many WBSNs as a user is enrolled. For instance, some years ago SixApart announced that WBSN users could reuse their social graphs anywhere. Indeed, [142] points out that much more work is required to implement mechanisms that really facilitate the reuse of profiles across various networking sites and applications. Nonetheless, despite that the reuse of identity data has been noticed, neither proposals have been developed in this regard, nor reusability of resources or access control policies has been mentioned.

3.3 Decentralized social applications

This Section describes the main existing decentralized WBSNs, Diaspora (Section 3.3.1) and Friendica (Section 3.3.2), which look for privacy and interoperability. Nonetheless, none of the WBSNs neither mention reusability, nor describe how they work to analyse if they could achieve reusability of some elements.

3.3.1 Diaspora

Diaspora[1] is a distributed social network based on a Peer-to-Peer (P2P) architecture in which peers store their personal resources in a particular host and leave them available for their contacts. Besides, cryptography can be applied to encrypt stored data.

The real focus of Diaspora is to achieve interoperability though not details of how to achieve it are provided. Indeed, this WBSN exclusively interacts with Diaspora servers, called pods. Moreover, either for not being an application trivial to install or for not being supported by successful platforms like Windows, Diaspora is far from being one of the most used WBSNs.

3.3.2 Friendica

Another decentralized WBSN is Friendica[2]. One of its main strengths is private data management. Besides, developers claim that Friendica supports friends from other applications like Facebook, Diaspora or Twitter.

Similar to Diaspora, this WBSN is not commonly used and no information about how it works is provided. Again, the low use of this WBSN may be caused by the installation process. Though developers are simplifying this process, it requires a significant effort in comparison with using famous WBSNs like Facebook or LinkedIn.

[1] https://joindiaspora.com/ , last access May 2014
[2] http://friendica.com/ , last access May 2014

3.4 Protocols

This Section describes protocols that facilitate the development of decentralized systems. Firstly, OpenID, a protocol to allow interoperable authentication (Section 3.4.1). Moreover, OAuth, a protocol that provides a method for clients to access server resources on behalf of a resource owner, is presented (Section 3.4.2). Lastly, UMA protocol, focused on access control decentralization, is introduced (Section 3.4.3)

3.4.1 OpenID

OpenID[3] is an open protocol that allows authentication. Users firstly register with an OpenID provider specifying their credentials, namely, a user name and a password. Then, this provider grants the user a URL that allows his authentication, together with a password as a complementary security measure. Using the provided URL and the password, users may authenticate themselves to a chosen service, which must implement OpenID, without the need of creating an account or being registered in the service.

In a nutshell, once in the possession of an OpenID URL, the protocol consists of the following steps: (1) the user contacts to the service (a relying party) that he wants to authenticate to and introduce his URL; (2) the service redirects the user to the user's OpenID provider which asks for the password; (3) the IdP verifies the password and, if the verification is successful, sends an authentication token to the service and redirects the user to the service. Consequently, identity data is interoperable and reusable because provided URLs allow users to authenticate themselves in any service running OpenID.

Despite the benefits that this standard provides, that is, simplifying authentication procedures and facilitating the interoperable authentication, it has not received a lot of attention. Currently, services like Flickr, MySpace, Google or Facebook,

[3]http://openid.net/ , last access May 2014

among others, implement this protocol. This situation may be produced due to the amount of security problems involved in the authentication process with OpenID, like phishing attacks [143]. Likewise, the fact that all requests pass through the IdP leads that this entity gets to know all access attempts.

3.4.2 OAuth

OAuth is an open protocol to allow secure authorization [144]. A service is allowed to access users' personal information that is stored in a service provider without disclosing any user credential to the service.

Assuming that a user stores photos in a service provider$_A$ and he desires using his photos in a particular service$_B$ without disclosing his credentials in service$_B$, the authentication of the user works as follows : (1) service$_B$ requests a request token to the service provider$_A$; (2) the service provider$_A$ grants a key to service$_B$; (3) service$_B$ directs the user to service provider$_A$; (4) service provider$_A$ obtains the users authorization for the request token and directs the user to service$_B$; (5) service$_B$ requests an access token to service provider$_A$ using the request token; (6) service provider$_A$ grants to service$_B$ the access token which can be used to access the protected resources.

OAuth is used in a well-known set of services like Google, Facebook, LinkedIn, Amazon, etc. Nonetheless, security issues are also at stake. For instance, users have to understand the meaning of authorizing a service to access data stored in a service provider, thereby avoiding the disclosure of more information than the one required for the requested service [145]. Moreover, apart from security, interoperability is another goal that this protocol tries to achieve but unsuccessfully. According to OAuth specification, some components are undefined, e.g. client registration, and then, services should be appropriately configured in respect to service providers to allow resources and identity data interoperability and reusability between different types of them. Indeed, in the lasted OAuth RFC the following is mentioned: "*How-*

ever, as a rich and highly extensible framework with many optional components, on its own, this specification is likely to produce a wide range of non-interoperable implementations".

3.4.3 User-Managed Access (UMA)

The UMA architecture and core protocol [1, 38], based on OAuth, achieves the following goals:

- Dedicated access relationship service: in different web domains, this service has to provide users with control over data-sharing and service-access relationships between online services hosting and accessing data.
- User-driven policies: users have to establish access policies over their data using their preferred managing tool.
- Support for claims: the establishment of access policies implies the attachment of properties that users, who want to access to a particular resource, must posses before the authorization can be granted.
- User management of access control: users are able to modify the conditions of access and terminate relationships easily.

On the other hand, UMA architecture consists of five entities (Figure 3.2):

- Requesting Party, who can be a corporation, a web user or any other legal person.
- Authorizing User (AU), who delegates access control from their chosen hosts to an Authorization Manager.
- Authorization Manager (AM), entity that acts on behalf of an AU and evaluates access requests made by a requester against applicable policies and issues access tokens if applicable. It plays the role of a PEP, a PDP and a Security Token Service,

- Host, web application that is used by an authorizing user to store and manage protected resources. It acts as a Policy Enforcement Point (PEP) because it carries out the delivery of data once presented the right access token.

- Requester, application that interacts with a host to get access to protected resources and interacts with an AM to obtain an access token.

Regarding the protocol, three steps are identified: (1) user introduces host to AM, which refers to the establishment of a trust relationship between the host and the AM. Besides, users specify policies in regard to uploaded resources; (2) requester gets access token from AM, that corresponds to the acquisition of a token to succeed in accessing a particular resource after delivering the appropriate claims; and (3) requester wields access token at host to gain access, which refers to the delivery of the requested token to obtain the requested resource. Particularly, claims are properties that must be satisfied to get a token. Current improvements are trying to reach a specific definition of claims, being IdPs in charge of their delivery.

Therefore, UMA provides key features to achieve resources and access control policies interoperability and reusability between different services because resources are stored in Hosts and access control policies in AMs, thus facilitating the access from different services to resources and access control policies.

Finally, it is noticeable that several UMA implementations[4] are developed, though not related to WBSNs. There are one commercial UMA authorization server and a total of three publicly available projects, particularly, the Fraunhofer AISEC project (which offers a client, an AM and a Host currently running), the OXAuth project (that facilitates the implementation of UMA for enterprise usages) and the SMART project (which involves the implementation of UMA together with sample applications). Besides, the first UMA interoperability testing[5] is being prepared to assess the interoperability between existing UMA implementations.

[4]https://kantarainitiative.org/confluence/display/uma/UMA+Implementations?src=context navchildmode , last access May 2014

[5]http://kantarainitiative.org/confluence/display/uma/UMA+Interoperability+Testing , last access May 2014

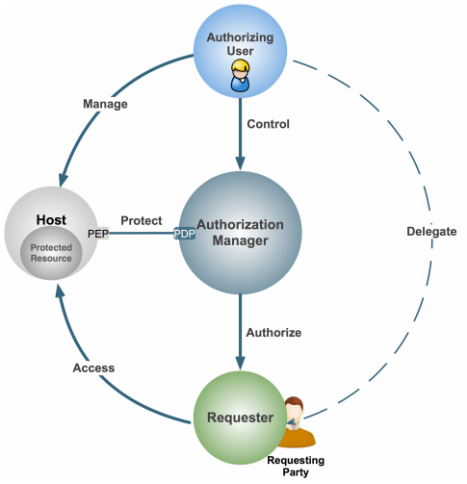


Figure 3.2: UMA interactions. Source: [1]

```
<foaf:Person rdf:about="#danbri" xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <foaf:name>Dan Brickley</foaf:name>
  <foaf:homepage rdf:resource="http://danbri.org/" />
  <foaf:openid rdf:resource="http://danbri.org/" />
  <foaf:img rdf:resource="/images/me.jpg" />
</foaf:Person>
```

Figure 3.3: FOAF example

3.5 Files format

This Section introduces the Friend-Of-A-Friend (FOAF) project (Section 3.5.1) and Microformats (Section 3.5.2), the main couple of file formats used to describe WBSN identity data, that is, profile and relationships data.

3.5.1 Friend-Of-A-Friend (FOAF)

FOAF is a project which provides a machine-readable ontology to describe people, things they create and do, and links between them [42]. It combines the use of the Resource Description Framework (RDF) and the Web Ontology Language (OWL). More specifically, the FOAF specification provides guidelines to structure and develop files in which personal data, such as name, phone, homepage, interests or photos or known users, like friends or relatives, are described. In particular, all defined tags are detailed in [146]. Nevertheless, FOAF is open, decentralized and extensible. Consequently, new enhancements are currently being performed. Figure

```
<link rel="profile" href="http://microformats.org/profile/hcard">
...
</head>
...
<ul class="vcard">
    <li class="fn">Joe Doe</li>
    <li class="nickname">Jo</li>
    <li class="org">The Example Company</li>
    <li class="tel">604-555-1234</li>
    <li><a class="url" href="http://example.com/">http://example.com/</a></li>
</ul>
```

Figure 3.4: Microformats example

3.3 presents an example of a person Dan Brickley with an image, a homepage and an openID identification.

Identified in [58], FOAF seems a promising approach in regard to the specification of users identity within the WBSNs' context. Multiple WBSNs, such as Twitter, and social applications, like Second Life, make use of it[6] [7].

3.5.2 Microformats

Microformats[8] allow adding information to web pages. It is used to describe some type of elements, e.g. a product, a person, a company, etc., and attached properties, e.g. a product has a brand, a name and a price. In general, Microformats use attribute *class* of HTML tags to assign short and descriptive names to entities, as well as to their properties. Several of them have been developed being hCard and hCalendar the ones that are ratified. For instance, applying hCard, Figure 3.4 presents the description of a user with a nick name, an organization and a telephone.

This approach is applied in several services. For instance, Facebook, Flickr and .Mac Webmail supports hCard. However, from the WBSN point of view, Microformats are not as user-friendly as FOAF because they are not particularly focused on people description.

[6] http://www.xul.fr/web-2.0.html, last access May 2014
[7] http://www.w3.org/wiki/FoafSites, last access May 2014
[8] http://microformats.org/, last access May 2014

Part III

Proposal

Chapter 4

SoNeUCON$_{ABC}$: an expressive usage control model for WBSNs and the enforcement mechanism

---

WBSNs consist of users who share data among other users through the establishment of relationships, emerging the necessity of mimicking daily life interactions [22]. The main goal is the development of expressive ACMs that facilitate the definition of any kind of user preferences from a private point of view.

There are a great amount of ACMs specially developed for the social networking field or that can be applied to this context but any of them offers the appropriate level of expressive power. As a starting point, concerning studies presented in Chapter 2, [93], [30], [29] and [27] are selected as the most expressive models.

Expressive power is mainly related to the management of relationships, users, objects and their respective attributes. Thus, an ABAC model seems to be an appealing model to work with. On the other hand, in contrast to current ACMs that focus on protecting resources until the access is granted and related to the sticky policy requirement, new developments require to manage access along the whole usage process [20]. For instance, undesirable copies or unnoticed dissemination of data should be avoided. In this regard, given that [29] and [27] are mainly based on relationships and put aside usage control, and [93, 30], based on $UCON_{ABC}$ [44], manage user, objects and consider usage control, the latter pair of works are

the chosen proposals to start up. Indeed, extending [29] or [27] would require, among other issues, the complex task of converting them into usage control models. Consequently, $UCON_{ABC}$ is extended by including relationship management and thus, Social Network $UCON_{ABC}$ ($SoNeUCON_{ABC}$), an expressive usage control model for WBSNs, along with its enforcement mechanism, is proposed. Note that $UCON_{ABC}$ has been chosen instead of [93] or [30] because of its maturity and scientific relevance. Besides, it is assumed that $SoNeUCON_{ABC}$ is complementary to [93] and [30] and, for instance, the addition of sessions and activities to $SoNeUCON_{ABC}$ is considered a matter of future work.

The work presented herein comprises in Section 4.1 the conceptualization of a WBSN. In Section 4.2 a formal description of $SoNeUCON_{ABC}$ is presented. Furthermore, an access control policy language using BNF notation is defined in Section 4.3. According to the proposed language, Section 4.4 details the enforcement mechanism. Finally, a high level architecture related to $SoNeUCON_{ABC}$ implementation is depicted and described in Section 4.5.

4.1 Conceptualization of WBSNs

Commonly, WBSNs are modelled as graphs, being Harary who, in 1953, applied graph theory regarding group behaviour, social pressure, cooperation, power and leadership [147]. Indeed, Harary is considered one of the pioneering of the application of graph theory to group behaviour.

More specifically, a graph is characterized by a huge quantity of entities, called nodes, and a vast quantity of connections between the nodes, called edges. In general terms, when modelling a WBSN as a graph, users correspond to the nodes and user relationships to the edges. This type of representation has been used by many authors in recent literature, being Carminati one of the most representative [17, 148, 28].

4.1.1 Resources

In a WBSN, the set of considered resources includes photos, videos, wall messages and personal messages that are private and directly written to a certain person or a group of people. Resources are a particular type of data which will be referred to as D.

Additionally, resources may be attached to multiple attributes that will be referred to as $ATD = \{atd_1, atd_2, ..., atd_{n_{ATD}}\}$ where n_{ATD} is the total number of attributes. Resources attributes can be classified in two groups. First group involves own features of resources such as the type, the creation time, the size and so on. Second group refers to any kind of characteristics that can be assigned to resources, for example the fact of being private, confidential or public, or the topic of the resource among others.

4.1.2 Actions

In a WBSN several actions can be performed on resources and identity data. The set of defined actions is denoted as AC. Main four actions that can be performed over resources and identity data are: *read*, equivalent to visualize any kind of content; *update*, equivalent to write down tags in videos or photos, or changing any commentary previously written; *insert* an element, equivalent to upload a photo or a video to the WBSN; and *delete* an element. Nonetheless, if needed, more actions can be considered.

4.1.3 Users

The set of users of a WBSN is identified herein as V. Although many types of users can be distinguished (recall Section 2.1.1), for simplicity but without losing generality and considering a single piece of data (resource or identity data), d_i, it will be distinguished the *administrator* of the data, i.e., $administrator(d_i)$, that is the user who administers the data access controls, and the *requesters*, i.e., the remaining

users of the WBSN that may request access to that piece of data. Therefore, all data administered by a user v_i is denoted as D_i, where $D_i \subseteq D$.

As it happened with resources, users may have a set of attributes which correspond to their profiles and are part of their identity data. This set of attributes is referred to as $ATV = \{atv_1, atv_2, ..., atv_{n_{ATV}}\}$ where n_{ATV} is the total number of users attributes. On the one hand, a user profile links each user with his/her nationality, age, music preferences and so on. On the other hand, there is other group of attributes, called user contextual attributes [149], that describe a user's personal mood, like happy, nervous and so on, or his/her current activity like eating, running, etc. A user's location is a particularly relevant user contextual attribute in WBSNs like Google Latitude, in which this information is the main data managed, or in WBSNs such as Facebook Places, in which user's location can be associated to data.

4.1.4 Relationships

In a WBSN, a user can usually accept and withdraw the establishment of relationships with other users of the WBSN. As mentioned above, relationships correspond to edges of the social graph which connects directly or indirectly pairs of users.

Direct relationships between users of a WBSN are identified herein as E. This set corresponds to contacts' information and it is part of users identity data. A set of attributes can be associated with a direct relationship. This set is denoted as $ATE_i = \{ate_1, ate_2, ..., ate_{n_{ATE}}\}$. The most important of these attributes are the relationship direction, namely, directional and bidirectional, and the relationship type, that is, the relationship semantic meaning which may be also called relationship role by some authors. Furthermore, users may be involved in several roles, e.g. being simultaneously "friend" and "co-worker" of a certain user.

Relationships may have other attributes such as creation time or history. Additionally, users may attach other attributes to their relationships such as a level of

trust [28] and a certain duration. Level of trust corresponds to a quantitative measure to determine the strength or weakness of a relationship. Concerning duration, it is the time during which the relationship remains valid.

Furthermore, two WBSN users may not be directly connected but indirectly, existing a path, P, connecting both nodes and composed of other users and their direct relationships. In this regard, the concept of user relationships and their attributes can be generalized to consider both direct and indirect types. The set of indirect relationship is then denoted as P and their attributes are derived from those involved in every $e_i \in P$.

4.1.5 Context

Context information is other aspect that may be considered when modeling a WBSN. The set of these features is denoted as CX. Dynamic, assorted and external features such as communication network status, service availability or other quality parameters can be involved here.

4.1.6 Summary of the conceptualization

The conceptualization provided above is summarized in this Section and depicted in Figure 4.1.

A WBSN is conceptualized as a directed multigraph, G, composed of the following elements: G:{ V, ATV, E, ATE, D, ATD, AC, CX}. V corresponds to the graph nodes, which represent the WBSN users and each of them has a set of associated attributes that can be derived using the mapping ATV. E are edges which represent direct relationships between pairs of users and each of them has a set of associated attributes derived with the mapping ATE. D refer to WBSN resources with have a set of attributes attached derived with the mapping ATD. Moreover, AC are the actions to be performed over objects. Finally, CX represents the system's context. Note that P, related to indirect relationships, is not included because this set of

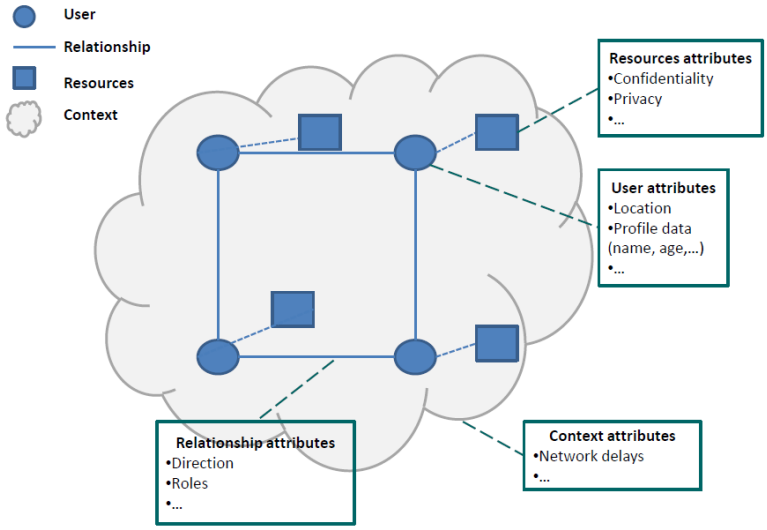


Figure 4.1: Web Based Social Network Conceptualization

relationships is constructed with E.

Finally, it should be noticed that as the conceptualization presented above is performed from a logical point of view, the way in which the storage is carried out, centralized like Facebook, decentralized such as Diaspora, encrypted or in plaintext, as well as, the way of specifying and managing relationships, are not particularly considered.

4.2 Formalization of SoNeUCON$_{ABC}$

Recalling that a WBSN can be defined as a graph, G, in which nodes (V) corresponds to users, edges (E) to relationships and resources (D) refer to elements that are somehow associated with users. Assuming this structure, the proposed model called $SoNeUCON_{ABC}$, mainly focuses on managing users (S), objects (O) and relationships (RT) attributes.

Contrary to $UCON_{ABC}$ which only manages direct relationships [150], the $SoNeUCON_{ABC}$ ACM extends the $UCON_{ABC}$ model by including a new independent entity, *relationships* (RT), and its attributes, *relationships attributes*

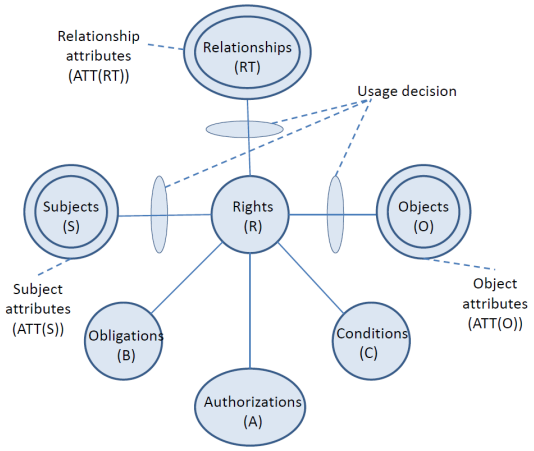


Figure 4.2: $SoNeUCON_{ABC}$

($ATT(RT)$). The new entity, RT, is composed of the sets of relationships (direct and indirect, as described below) between pairs of users. The remaining entities, attributes and functions considered in the $UCON_{ABC}$ model are also considered in the $SoNeUCON_{ABC}$ model. Access control is now managed through the establishment of policies defined over authorizations (A) built over $ATT(S)$, $ATT(O)$, $ATT(RT)$ and rights (R), obligations (B) and conditions (C). Note that graph terminology is applied according to [151] and that elements of the WBSN conceptualization can then be related to those in $SoNeUCON_{ABC}$ as follows:

- *Subjects* (S) are the WBSN users (V) who play the role of requesters. User attributes, referred to as users profiles, can be derived with the mapping $vAT : S \longrightarrow ATT(S)$. Notice that vAT includes multiple individual mappings such that $vAT_i : S \longrightarrow ATT(S)_i$. For example, $vAT_1 : S \longrightarrow AGE$ where AGE contains all possible values for the AGE attribute.

- *Objects* (O) are WBSN resources (D) that refer to as photos, videos, wall messages and personal messages. Additionally, they have attached a set of object attributes that can be derived with the mapping $dAT : O \longrightarrow ATT(O)$. For instance, a possible individual mapping is $dAT_1 : O \longrightarrow TITLE$.

- *Relationships* (RT) represent the set of existing relations between the users of

the WBSN which is part of users identity data. Under the approach taken in this work, given a request (s, o, r) (where s is the requester, o the requested object and r the requested right over o), a specific set of relationships rt is considered to manage access control. In particular, rt is defined as the set of relations that exist between the administrator a of the requested object o and the requester s.

Consider that a *path* p in G is a sequence of alternating vertices $v_i \in V$ and edges $e_i \in E$ such that $p_i = (v_{i_0}, e_{i_1}, v_{i_1}, \ldots, v_{i_{j-1}}, e_{i_j}, v_{i_j}, \ldots e_{i_k}, v_{i_k})$ where i is the path identifier, j the node order and k the length of the path. Let $\mathcal{P}$ be the set of all simple strong paths in G. A path p_i is said to be *simple* if no node occurs more than once, i.e., $\forall i_q, i_r$ where $q, r \in \{0, \ldots, k\}$, if $i_q \neq i_r$, then $v_{i_q} \neq v_{i_r}$. A path p_i is said to be *strong* if for all nodes v_{i_j} in the path, except for the initial node v_{i_0}, there exists an edge e_{i_j} in the path such that $e_{i_j} = (v_{i_{j-1}}, v_{i_j})$ that is, an edge that links forwards both nodes $v_{i_{j-1}}$ and v_{i_j} (in the direction from $v_{i_{j-1}}$ to v_{i_j}). In Figure 4.3, a simple strong path may be found between v_8 and v_1 using the edges referred to as e_{13}, e_{21} and e_7. The path will be $(v_8, e_{13}, v_5, e_{21}, v_3, e_7, v_1)$. Note that in the case of the figure the number in the subscripts is just an identifier of the specific node or edge.

Let V_{p_i} and E_{p_i} be respectively the node set and edge set of path p_i. Then, p_i^*, the *enriched path* of p_i, is built by adding to E_{p_i} the edges of G that link, either forwards or backwards, two consecutive nodes of path p_i. Thus, $\mathcal{P}^*$ is the set of enriched paths p_i^* built from paths in $\mathcal{P}$. Enriched paths are represented as $p_i^* = (v_{i_0}, (e_{i_1}^1, \ldots; e_{i_1}^2, \ldots), v_{i_1}, \ldots, v_{i_{j-1}}, (e_{i_1}^{l_j}, \ldots; e_{i_2}^{l_j}, \ldots), v_{i_j}, \ldots, v_{i_{k-1}}, (e_{i_k}^1, \ldots; e_{i_k}^2, \ldots), v_{i_k})$, being l_j the multiplicity of edges between nodes $v_{i_{j-1}}$ and v_{i_j} and k the length of the path. Note that edges that link two consecutive nodes of the path have been classified in two groups (separated by a semicolon), that contain respectively the forward and backward edges. Symbol $\varnothing$ indicates that no edge exists in that direction. For instance, in

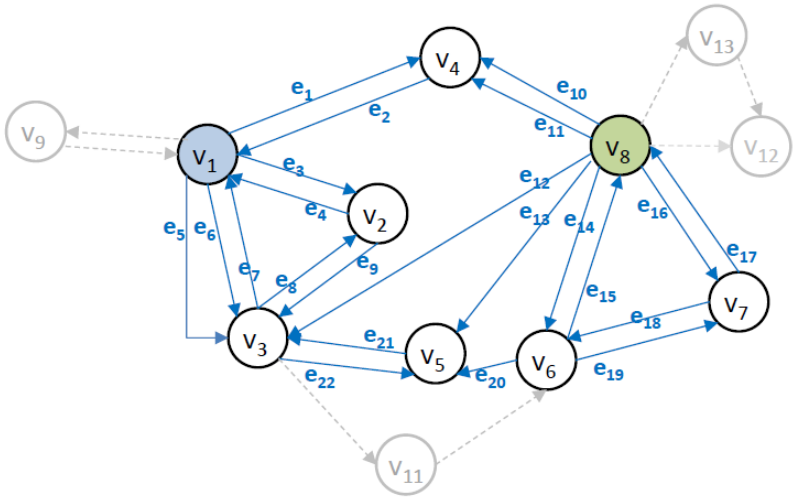


Figure 4.3: Relationships example, $G_{RT[v_8,v_1]}$

Figure 4.3, given the set of nodes $\{v_8; v_5; v_3; v_1\}$, p_1^* is an enriched path defined as $(v_8, (e_{13}; \varnothing), v_5, (e_{21}; e_{22}), v_3, (e_7; e_6, e_5), v_1)$. Note that edge e_{12} is not included because although it links two nodes of path p_1^*, these nodes are not consecutive. This example highlights that enriched paths facilitates expressive power management because they allow navigating along all the relationships between a pair of users, no matter if they are direct or indirect.

Let v_a and v_s be respectively the nodes representing the administrator a of the requested object o and the requester s. Then rt is built as the set of all enriched paths, from $\mathcal{P}^*$, that allow reaching node v_s from node v_a. It is considered that each enriched path in rt is a relationship between v_a and v_s. In the example shown in Figure 4.3, rt for $v_a = v_8$ and $v_s = v_1$ is formed by the vertex set $V_{rt} = \{v_8, v_4, v_2, v_7, v_6, v_5, v_3, v_1\}$ and the edge set E_{rt} is composed by all continuous arrows.

Attributes of the relationships between v_a and v_s are derived from the attributes of the direct relationships that compose them. The set of attribute values associated to direct relationships is denoted as $ATT(E)$ and can be derived with the mapping $eAT : E \longrightarrow ATT(E)$. As in the case of the

other attribute mappings, eAT is composed by multiple individual mappings; for example, $eAT_i : E \longrightarrow ROLE$, where $ROLE$ contains all possible values for the $ROLE$ attribute. As enriched paths contain one or more edges, the concatenation of the attributes of these edges constitute the set of attributes of the enriched path and they can be derived with the mapping $pAT : P^* \longrightarrow ATT(P)$ with $ATT(P) = ATT(E)^{n_e}$ and n_e the number of edges in the considered enriched path. The structure of the enriched path (number of edges between two consecutive nodes and their direction) needs to be kept, therefore, the set of its attributes may be represented following the notation used to represent an enriched path but excluding the nodes. Consider, for example, the enriched path p_1 such that $(v_8, (e_{10}, e_{11}; \varnothing), v_4, (e_2; e_1), v_1)$. Let's assume that only edge attributes of $ROLE$ and $TRUST$ are considered, and that their values for the edges in p_1 are $eAT(e_{10}) = (friend, 3)$, $eAT(e_{11}) = (colleage, 2)$, $eAT(e_2) = (friend, 4)$ and $eAT(e_1) = (friend, 3)$. Then, the attributes of the enriched path p_1 will be represented as $pAT(p_1) = (((friend, 3), (colleage, 2); \varnothing), ((friend, 4); (friend, 3)))$. Therefore, a set of relationships, rt, has also associated a set of attributes $ATT(RT)$, which can be derived with the mapping $rtAT : rt \longrightarrow ATT(RT)$ and built similarly from the attributes of its enriched paths.

Note that although rt is defined as the set of enriched paths between v_a and v_s, the relationships information could be condensed in the subgraph induced from the nodes involved in those relationships.

- *Rights* (R) refer to the actions (AC) that can be performed over WBSN resources and identity data such as read, update or delete.

- *Authorizations* (A) are the rules defined as functional predicates over $ATT(S)$, $ATT(O)$, $ATT(RT)$ and R, that have to be satisfied, before or while the usage process, in order to grant a right to a subject on an object.

- *Obligations* (B) refers to requirements that users have to satisfy before or while the usage process.

- *Conditions* (C) correspond to requirements regarding the system or the environment status that have to be satisfied before or while the usage process.

Lastly, it is remarkable that in this model the unique data managed are the attributes of the requester, of the administrator of the requested object and of the relationships between the administrator and the requester. Therefore, attributes of the rest of nodes involved in indirect relationships remain hidden when access control is enforced.

4.3 Access control policies

A request is expressed as $\{s, o, r\}$. In $SoNeUCON_{ABC}$, access control policies (ρ) are defined in terms of authorization A (predicates over $ATT(S)$, $ATT(O)$, $ATT(RT)$ and R), obligations B and conditions C. As illustrated next, the requested $r \in R$ over $o \in O$ is granted if the values of $ATT(S)$ of the requester, $ATT(O)$ of the requested object and $ATT(RT)$ the set of relationships between the administrator and the requester satisfy the appropriate ρ. Note that although the general definition of access control policies is based on $ATT(RT)$, given that $ATT(RT)$ is built from $ATT(E)$, this last set of attributes is the one directly managed during policies specification and enforcement. Therefore, the key matter is the definition of the set of attributes types regarding $ATT(S)$, $ATT(O)$ and $ATT(E)$, detailing possible operators, as well as, the way of constructing policies.

4.3.1 Policy attributes

Attribute types can be classified as follows:

- Boolean ($\mathcal{B}$). These attributes can take a pair of values. For instance, the subject attribute *married* can take values "Yes" or "No".

- Free-valued ($\mathcal{FV}$). These attributes can take a value from a set of multiple possibilities. This attribute type is divided in two groups:
  - Numeric ($\mathcal{M}$): it refers to attributes which have numeric values. For example, the subject attribute *age* may take values from 15 to 99.
  - String ($\mathcal{S}$): it corresponds to attributes whose values are strings of characters. A key example is the direct relationship attribute *role* which takes values "friend", "colleague", "relative" and so on.
- Data structures ($\mathcal{D}$): it refers to structures in which numeric, string and boolean attributes can be combined. For instance: the direct relationship attribute *creationTime* takes values following the pattern "dd/ mm/yyyy-hh:mm", where *dd* refers to the day, *mm* refers to the month, *yyyy* refers to the year, *hh* refers to the hour and *mm* refers to the minute when the relationship was established. Moreover, notice that the construction of multivalued attributes can be created as a data structure, e.g., a list.

4.3.2 Policy operators

Regarding attributes management, the following set of operators can be applied:

1. *Logical operators* $(\mathcal{L})$ ::= $\wedge|\vee|\neg$. Operator $\wedge$ represents a logic AND, operator $\vee$ refers to a logic OR and operator $\neg$ represents negation. Operators $\wedge$ and $\vee$ can be applied to two elements that can be attributes of boolean type or boolean expressions result of applying a functional or complex operator (see below). Operator $\neg$ can be applied to one of these elements.
2. *Functional operators* $(\mathcal{F})$ ::=$< \mid > \mid \leqslant \mid \geqslant \mid =$. These operators are applied to two attributes of numeric or string type. For example, given the subject attribute *age*, the age of a pair of users, v_1 and v_2, can be compared by building the expression $age(v_1) < age(v_2)$ to identify if v_1 is younger than v_2. The result of applying these operators is a boolean value. Note that a concrete

specification of string types management should be defined according to each particular context.

3. *Complex operators* ($\mathcal{X}$). These operators are an open set based on the combination of logic and functional operators and they are applied to data structures. For instance, assuming the existence of a list of attribute values such that $\{att_iValue_1, att_iValue_2, ..., att_iValue_n\}$, $x_i \in \mathcal{X}$ can be defined as an operator which goes recursively through the list until identifying a particular attribute value. The result should be also a boolean value.

4.3.3 Policy construction

Concerning the above operators, constraints and policy attributes, each policy (ρ) in $SoNeUCON_{ABC}$ model is generally depicted as $\rho(A; \partial_b; \partial_c)$, that is, $\rho(\rho_s; \rho_o; \rho_{rt}; r; \partial_b; \partial_c)$.

More specifically, ρ_s, ρ_o and ρ_{rt} correspond to predicates ($\alpha_{att(w)_i}$) built using operators applied to the attributes of a subject, $ATT(S)$, an object, $ATT(O)$, and the set of relationships between the subject and the administrator of the requested object, $ATT(E)$, where w in $\alpha_{att(w)_i}$ can take three possible values – s, o or e –, to indicate the set of attributes that the attribute used in the expression belongs to. Moreover, ∂_b and ∂_c refer to sets of obligations and conditions. Policies are formally described through BNF notation [152]. For the sake of simplicity, when a specific mapping on an entity should be used to obtain the value of an attribute (e.g., $age(s)$, $title(o)$, or $trust(e_i)$), only the name of the mapping will be used (i.e., *age*, *title*, or *trust*), as the policy construction allows to clearly identify the entity to which the mapping should be applied in each case.

- Regarding ρ_s and ρ_o, ρ_s (analogously ρ_o) is defined as follows:
  - $\rho_s ::= (\text{`}\varnothing\text{'} | ([\text{`}\neg\text{'}]\ \alpha_{att(s)_1})\ \{\ \text{`}\wedge\text{'} | \text{`}\vee\text{'}\ ([\text{`}\neg\text{'}]\ \alpha_{att(s)_j})\}^*)$, which means that predicates applied over subjects attributes can be connected by disjunctives and/ or conjunctives and/or negatives ($\alpha_{att(s)_i}$) For instance:

$\rho_s = (age > 18 \wedge student = uc3m)$

- By contrast, ρ_{rt} is composed of predicates built over $ATT(RT)$. It consists of a list of three elements: the first one (σ) corresponds to the set of conditions (expressed as paths, ψ) that enriched paths in rt must satisfy; the second one (ϖ) refers to the number of times that the only path σ involves, should exist in rt; and the last one (δ) corresponds to the number of nodes involved in a clique, being σ composed of a single forward direct relationship. Indeed, δ is directly related to the number of different bidirectional enriched paths that exist in a clique with the formula $(\sum_{K=1}^{N} P(K,N)+1)$ where $N = \delta - 2$ (N is the number of members of the clique excluding v_a and v_s) and $P(K,N)$ refers to the number of K-permutations in a set of N elements.

  According to the previous description of ρ_{rt} elements, the proposed policy language allows the definition of policies with cliques (applying δ) and policies that involve conditions regarding multiple paths (applying σ), multiple instances of one particular path (applying ϖ) or multiple types of different paths (applying multiple and analogous ψ in σ).

  - ρ_{rt} ::=‘$\varnothing$’ | $(\sigma,\ \varpi,\ \delta)$

  - σ ::= $(\psi\ \{\text{‘}\wedge\text{’}|\text{‘}\vee\text{’}\ \psi\}^*)$, which means that σ is composed of the disjunction and/ or the conjunction of multiple paths ψ.

    * ψ ::= $(\tau\{;\tau\}^*)$, which means that each path ψ is composed of a set τ of direct relationships at different hops, being symbol “;” applied to distinguishing hops. The amount of hops in a path can be calculated in respect to the amount of “;” plus one.

      · τ ::= $fert|bert|\text{‘}\varnothing\text{’}\ \{\text{‘}\wedge\text{’}|\text{‘}\vee\text{’}\ fert|bert|\text{‘}\varnothing\text{’}\}^*$, which means that τ is composed of the disjunction and/ or the conjunction of forward $fert$ and backward $bert$ relationships.

· $fert$ ::= ([‘¬’] $\alpha_{att(e)_i}$ {‘∧’|‘∨’ [‘¬’] $\alpha_{att(e)_i}$ }*), which means that each direct forward relationship $fert$ is built over the disjunction and/ or the conjunction and/or the negation of predicates applied over relationship attributes ($\alpha_{att(e)_i}$).

· $bert$ ::= $-fert$

For example: ρ_{rt} = ((((*role* = *friend*) ∧ −(*role* = *friend*)) ∧ ((*role* = *friend*); (*role* = *friend*))), ∅, ∅) refers to the existence of a pair of paths. One of them corresponds to a forward and a backward friendship relationship. The other one corresponds to a friend of a friend forward relationship.

– ϖ ::=‘∅’|n, where $n \in \mathbb{Z}$ and $n > 2$. Note that in case ϖ takes value n, σ must be composed of a single path ψ to identify n occurrences of such path. Otherwise ϖ is ∅. For instance: ρ_{rt} = {{{{*role* = *friend*}; {*role* = *relative* ∧ *creationYear* > 2010}}}, 2, ∅} expresses the necessity of existing, at least, a pair of paths of length two where the first hop involves a forward friendship relationship and the second hop involves a forward relative relationship which must have been created after the year 2010.

– δ ::=‘∅’|n, where $n \in \mathbb{Z}$ and $n > 0$. Note that if δ takes a value distinct from ∅, then σ must correspond to a single path ψ composed of a forward relationship. Besides, the existence of a backward relationship of the same type as the forward one is assumed. Otherwise δ is ∅. For instance: ρ_{rt} = ((((*role* = *friend* ∧ *trust* = *high*))), ∅, 3) corresponds to the existence of a clique of three users where the relationships between each pair of users are highly trusted and have the *role friend*.

- r ::= $write|read|....$
- ∂_b ::= (‘∅’ | {$abligation_n$}*).
- ∂_c ::= (‘∅’ | {$condition_n$}*).

Then, a right (r) to an object (o), whose attributes satisfy ρ_o, is allowed if the attributes of the requester (s) satisfy ρ_s, the set of relationships between the administrator of o and the requester, directly or indirectly, satisfy ρ_{rt}, and ∂_b and ∂_c are also satisfied: $allowed(s, o, r) \Rightarrow \rho(\rho_s; \rho_o; \rho_{rt}; r; \partial_b; \partial_c)$. Notice that components of ρ, except for r, can be empty ($\varnothing$). This issue can be used to specify public policies or establish undetermined object, subject or relationship conditions. For instance: $\rho = (\varnothing; (title = party); (((\varnothing; \varnothing))), \varnothing, \varnothing); read; \varnothing; \varnothing)$ expresses that objects entitled "party" can be read by users whom the administrator has some kind of relationship with, as long as users are located at a maximum length of 2 hops.

On the other hand, concerning *mutability* and *continuity* management, policies can be analogously defined for pre and on-going usage processes. Specifically, they are defined as $\rho_{pre}(\rho_s; \rho_o; \rho_{rt}; r; \partial_b; \partial_c)$ and $\rho_{on-going}(\rho_s; \rho_o; \rho_{rt}; r; \partial_b; \partial_c)$. Then, as long as attributes change and regarding the stage of the usage process, the appropriate access control policies are evaluated. Nonetheless, the unpredictability of conditions and obligations definition has to be discussed. For instance, in the WBSN Badoo, an obligation exists such that three photos should be uploaded to the personal profile before accessing to other users' albums. On the other hand, a devised obligation may require to have five contacts before accessing to other users' profiles. By contrast, a possible condition is referred to the system's load capacity (e.g. free or busy), that is, accesses are rejected until the system is free. These examples illustrate the large set of possible conditions and obligations that can exist. Consequently, such variability together with the aim of getting as much flexibility as possible, support the inappropriateness of providing a concrete specification of both types of elements herein.

4.4 Access control enforcement

Access control enforcement is carried out through the execution of a set of functions which, in this proposal, are linked to the proposed policy language.

Once a request $\{s, o, r\}$ is received, stored policies are retrieved. A pair of alternatives to enforce access control are identified. The first one is based on searching for enriched paths between the administrator a of the requested object o and the requester s throughout the whole WBSN graph (G), and verifying the policies during the process. On the other hand, the second approach first builds the set rt. Afterwards, policies are verified considering this set. Given the difficulty in accurately measuring the complexity of the first alternative, the second alternative is adopted herein. This alternative allows the reuse of rt while verifying policies, as well as the calculation of the upper bound of the computational complexity concerning access control enforcement.

The enforcement process is divided in four main activities, see diagram of Figure 4.4, namely, 1) the enforcement setup, 2) the verification of policy elements which is composed of six activities related to the evaluation of elements involve in policies, 3) the construction of rt between the administrator a and the requester s and 4) the usage control process. The first activity consists of retrieving policies of the administrator. Then, per each policy ρ, the right r, ρ_s and ρ_o are evaluated. Subsequently, rt is constructed (if required) and ρ_{rt} is evaluated. The next activities correspond to the evaluation of conditions and obligations (∂_c and ∂_o respectively). Finally, access control is managed along the whole usage process. Note that the set of ρ_{pre} is the first one evaluated. Besides, note that the evaluation order of elements involved in each policy ρ can be altered due to efficiency reasons and, for instance, conditions can be evaluated before ρ_{rt}.

More specifically, the work-flow of the enforcement process in terms of applied activities and actions involved in them is depicted in Figures 4.5 and 4.6. Along this section, actions correspond to the developed functions and they are pointed out in square brackets. The enforcement setup involves the retrieval of access control policies of the administrator a of the requested object o, to start the evaluation of each of them [$FindSubjectPolicies$].

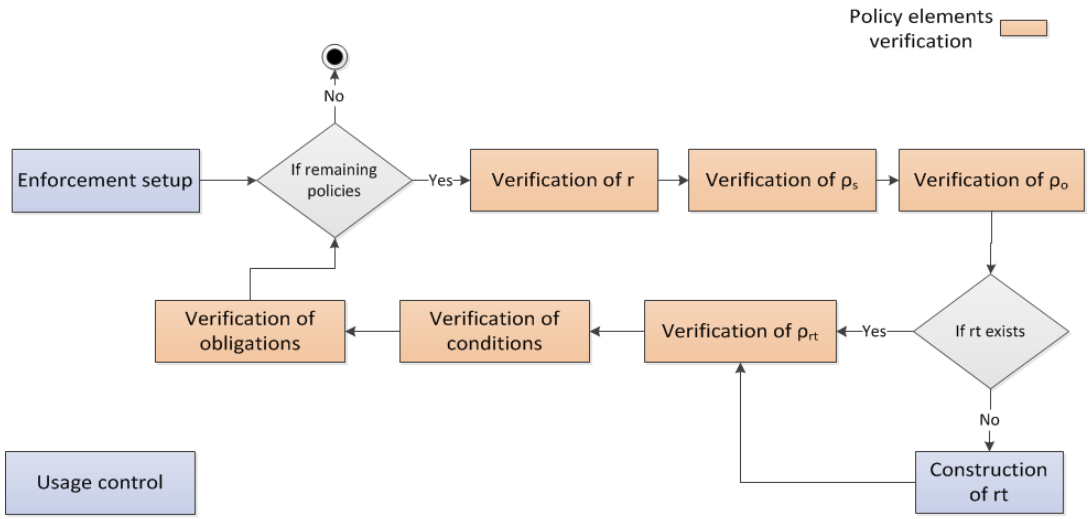


Figure 4.4: Activity diagram of the enforcement process.

Regarding the construction of rt, it consists of the set of enriched paths between the administrator a and the requester s [$CreateRT$]. This structure is recursively built identifying contacts of the administrator a, at different hops, from whom the requester s is reached and storing attached enriched paths [$GetNumContacts/$ $GetConnectedUser$]. The recursion is performed and rt constructed until path length is 6. This upper limit is established due to theoretical studies [153]. Note that $ATT(S)$ of intermediate nodes are not retrieved and evaluated. An identifier of these intermediates nodes is only necessary to jump from one node to another until constructing rt. Besides, identifiers do not have to disclose users' information.

Another activity is the evaluation of policies. Per each policy ρ, elements involved in it are evaluated, namely, subject, objects and relationships attributes (involved in ρ_s, ρ_o and ρ_{rt} respectively), the established right r, conditions (∂_c) and obligations (∂_c) [$Match$]. Firstly, the requested right r is evaluated against the right $r \in \rho$ [$MatchR$]. Next, it is verified if predicates ($\alpha_{att(s)_i}$) involved in ρ_s and built over different types of $ATT(S)$ (boolean $\mathcal{B}$, free-values $\mathcal{FV}$ or data structures $\mathcal{D}$), match attributes of the requester s [$VerifyDAttTypes/$ $VerifyFVAttTypes/$ $VerifyBAttTypes$]. Subsequently, the same process is performed for ρ_o and at-

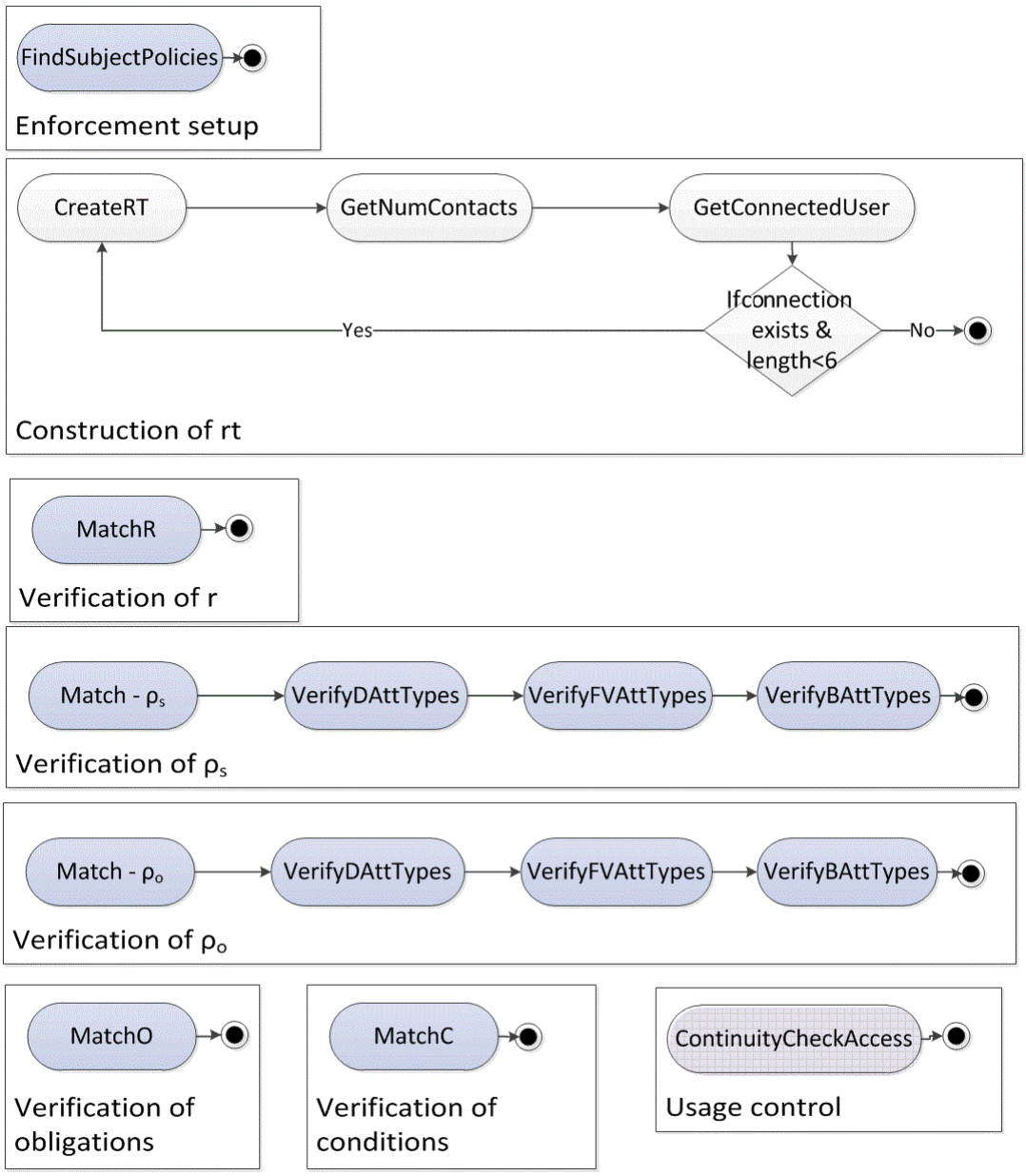


Figure 4.5: Enforcement process work-flow.

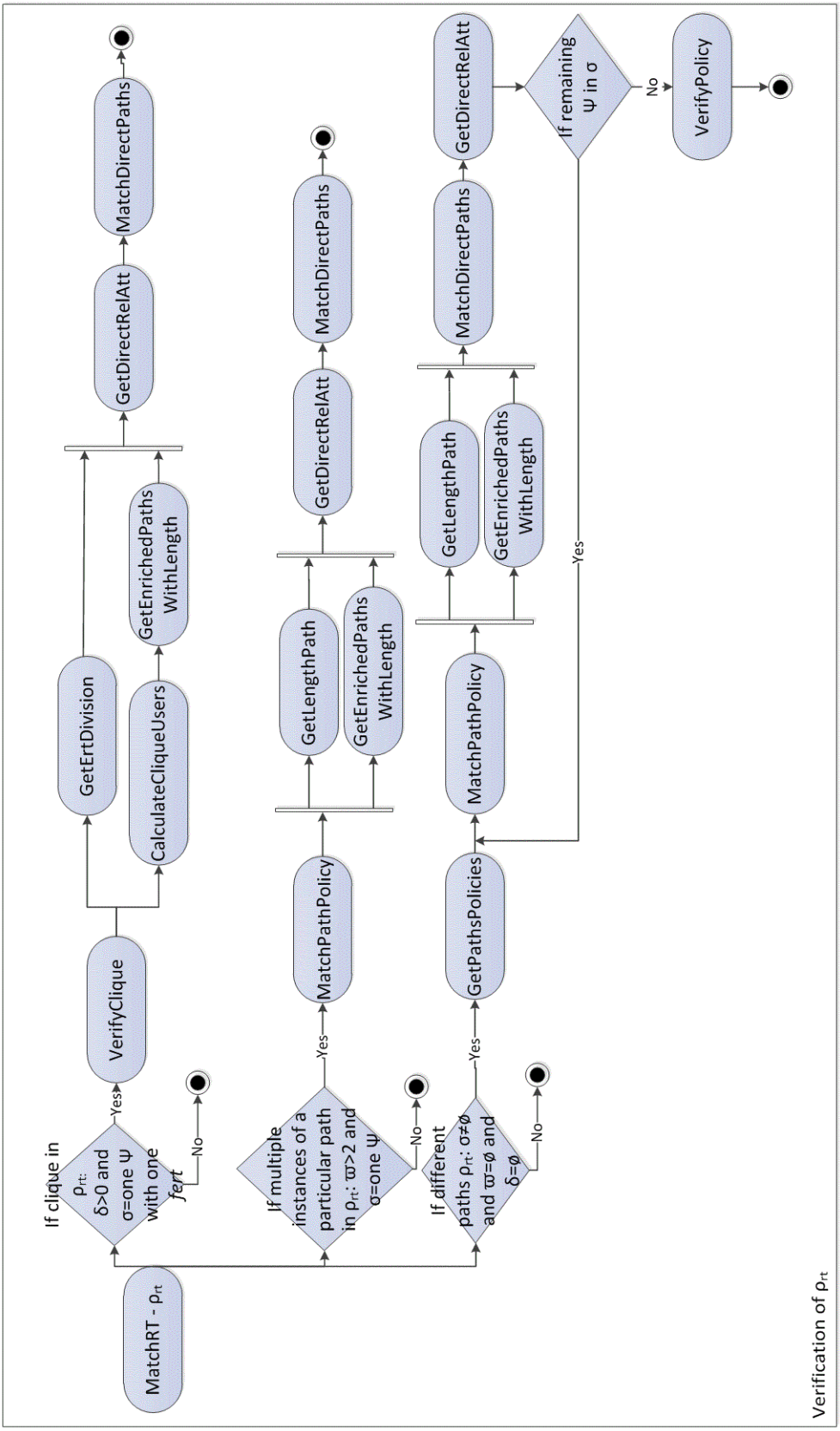


Figure 4.6: Enforcement process work-flow of the verification of ρ_{rt}.

tributes of the requested object o.

Thirdly, the evaluation of ρ_{rt} is carried out in three different steps [$MatchRT$], see Figure 4.6. The primary step is the validation of cliques when $\delta > 0$ and σ consists of a single path ψ with just a forward direct relationship $fert$ in τ [$VerifyClique$]. If a clique is evaluated, the lengths of all possible enriched paths in the clique between the administrator a and the requester s are initially calculated [$CalculateCliquePaths$]. More specifically, these lengths are calculated from the formula $\sum_{K=1}^{N} P(K,N) + 1$ (recall Section 4.3.3) where K-permutations in a set of N elements should be performed, being N the number of members of the clique distinct from the administrator a and the requester s. Then, all enriched paths in rt whose length corresponds to the calculated ones, are stored [$GetEnrichedPathsWithLength$]. Simultaneously, the forward relationship $fert$ from σ is retrieved [$GetErtDivision$]. Finally, stored paths are processed identifying forward and a backward relationship at each hop [$GetDirectRelAtt$] and verifying if these relationships satisfy attribute predicates of $fert$ [$MatchDirectPaths$].

The second step in the evaluation of ρ_{rt} is the verification of multiple instances of a particular type of path in rt [$MatchPathPolicy$]. This issue is verified when $\varpi > 2$ and σ consists of a single path ψ. After identifying the length of path ψ [$GetLengthPath$], enriched paths of rt are processed storing those with such length [$GetEnrichedPathsWithLength$]. Then, ψ is processed retrieving all direct forward and backward relationships in each hop, that is, τ [$GetDirectRelAtt$]. Similarly, regarding stored paths of rt, forward and backward relationships in each hop are also retrieved [$GetDirectRelAtt$]. The last part is the evaluation of retrieved backward and forward relationships of ψ against those of paths of rt [$MatchDirectPaths$].

The last step in the verification of ρ_{rt} consists of evaluating paths of different types, that is, when σ is the only element in ρ_{rt} different from $\varnothing$. It is equivalent to the second step but individually processing every ψ in σ [$GetPathsPolicies$]. Once

all paths ψ are evaluated, it is finally verified that enriched paths of rt satisfy σ, that is, it is verified that all paths are connected according to operators ($\wedge$ and/ or $\vee$) σ involves [$VerifyPolicy$].

Policies evaluation finishes with the verification of obligations (∂_b) and conditions (∂_c) [$MatchO/\ MatchC$]. However, given their variety, ∂_b and ∂_c, enforcement should be attached to each particular implementation and thus, left for systems' developers.

Finally, usage control has to be taken into account. Each time an attribute or a policy is somehow updated, the enforcement process is repeated applying the appropriate rt previously constructed or re-constructing a new if the stored rt is affected by produced changes [$ContinuityCheckAccess$]. Moreover, it is noticeable that $\rho_{on-going}$ management is attached to particular implementations. Therefore, usage control has to be integrated within an entity (such as, the PDP and/or the PEP recall Section 2.1) who adequately manages data requests and changed elements.

More details regarding applied functions are presented in Appendix D.1, where, based on [154], the concrete specification of every function is depicted and explained. Note that, for simplicity reasons, auxiliary functions such as $GetFirstNode$, are not pointed out herein but their use is detailed in the Appendix.

4.5 $SoNeUCON_{ABC}$ high level architecture

$SoNeUCON_{ABC}$ is an ACM and then, multiple architectures can be developed in its regard. The architecture depicted in Figure 4.7 is the one proposed herein. For simplicity reasons, it is centralized and access control management is enforced within each WBSN. Nonetheless, a decentralized architecture considering a simplified version of $SoNeUCON_{ABC}$ is also proposed in this Thesis and it is described in Chapters 7 and 8.

WBSNs store and manage users, objects and relationships. WBSN users upload

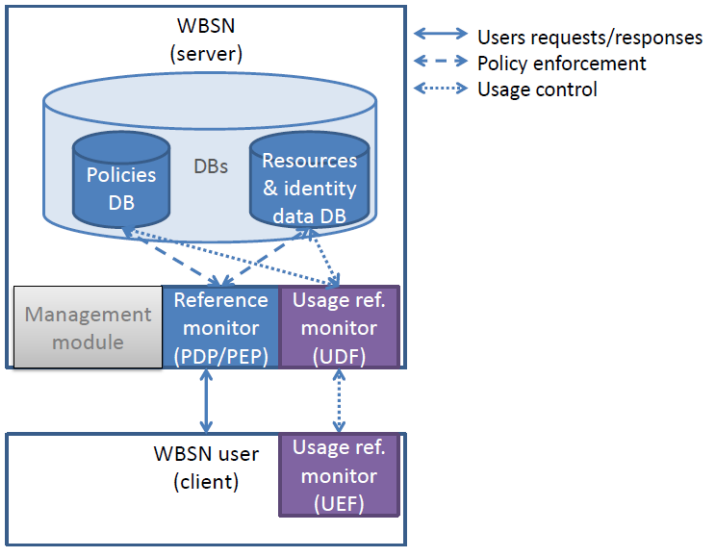


Figure 4.7: $SoNeUCON_{ABC}$ high level architecture

objects, referred to as resources, define access control policies, establish identity data, namely, their contacts' information and their personal attributes, and request rights over other users' resources or identity data. More specifically, the high level architecture consists of the following four components:

- *Data bases (DBs)*: there are as many DBs as each WBSN requires. Mainly, stored data refers to access control policies, resources and identity data
- *Management module*: this module allows to perform administrative operations (detailed in Chapter 5), such as the creation and update of access control policies.
- *Reference monitor*: as described in Section 2.1, the reference monitor is the core component of access control systems and it is responsible for the enforcement process. Given users requests, it verifies requests versus access control policies and grants or denies access accordingly. In what concerns ρ_{pre}, the reference monitor is located at the server-side, that is, where access control policies are stored and evaluated but it may depend on particular implementations.

- *Usage reference monitor*: enforcement in regard to $\rho_{on-going}$ requires the introduction of a Usage Decision Facility (UDF) and a Usage Enforcement Facility (UEF) (recall Section 2.5.3.1) which compose an entity called herein usage reference monitor. Although it must be studied in respect to particular implementations, UDF may be located at the server-side to detect attributes changes and UEF may be located at the client-side (i.e. in the browser) to enforce access control along the usage period. As a matter to notice, developers must be aware of the most critical web application security risks ([155]) in order to face them.

The relation between $SoNeUCON_{ABC}$ entities and above components is clear. DBs store information related to users (S), their resources (O and $ATT(O)$), identity data (users' profiles, $ATT(S)$, and contacts' information, E and $ATT(E)$) and access control policies (A), the reference monitor manages access control concerning ρ_{pre} and the usage reference monitor takes part in the usage control process in what concerns $\rho_{on-going}$.

4.6 Summary of the chapter

In this Chapter, an expressive ACM, called $SoNeUCON_{ABC}$, has been presented together with a policy language, the attached enforcement functions and its high level architecture. This model achieves fine-grained access control management in a privacy friendly way, such that subject attributes of nodes involved in the relationships between the administrator and the requester of an object remain hidden.

Chapter 5

SoNeUCON$_{ADM}$: administrative model for SoNeUCON$_{ABC}$

Some of the uploaded data to WBSNs are personal data and they are in multiple cases out of control. A careful supervision and management of all WBSNs data is a demanding and a challenging necessity. At a primary step, $SoNeUCON_{ABC}$, an expressive usage control model that allows fine-grained access control management has been proposed in Chapter 4. However, the identification and specification of administrative tasks is the following step. In this Chapter $SoNeUCON_{ADM}$, an administrative model for $SoNeUCON_{ABC}$, is presented.

This Chapter is structured as follows. Section 5.1 introduces administrative features, particularly, tasks and rights. In Section 5.2 $SoNeUCON_{ADM}$ is described. Lastly, Section 5.3, according to $SoNeUCON_{ABC}$ high level architecture (see Section 4.5), describes components involved in administrative management.

5.1 Towards administration

Prior to the description of how administration is performed in $SoNeUCON_{ADM}$, administrative tasks to address (Section 5.1.1) and the available rights to manage (Section 5.1.2) are detailed in the following Sections.

5.1.1 Administrative tasks

Studied in Section 2.6.1, administration involves multiple tasks. They can be classified in a couple of groups regarding tasks related to:

- *The identification of who is involved in administrative issues.* These tasks refer to who manages access control policies, who associates policies with resources and identity data and who manages revocation and delegation.
- *The definition of how administrative issues are performed.* These tasks correspond to how policies are associated with resources and identity data, how resources and identity data are associated with their owners and how revocation and delegation are managed.

5.1.2 Rights management

Two types of rights are differentiated, *use rights* and *administrative rights*. The former ones, which are referred in $SoNeUCON_{UCON}$ to as Rights (R), are based on operations that can be performed with objects, e.g. read, and operations that can be carried out over objects, e.g. tag, move, copy, etc. By contrast, *administrative rights* (AR) refer to the management of elements involved in the access control decision process, as well as the management of delegation and revocation.

5.2 $SoNeUCON_{ADM}$ definition

Users enrolled in a WBSN become owners of uploaded resources, established identity data (mainly profile data) and defined access control policies (recall Chapter 2). Thus, $SoNeUCON_{ADM}$ is based on **ownership**, such that owned elements are managed by their owners. Particularly, administrative objects (AO) refer to elements involved in the access control decision process, namely, managed subjects (S) and their attributes ($ATT(S)$), objects (O) and their attributes ($ATT(O)$), direct relationships (E) and their attributes ($ATT(E)$) and access control policies

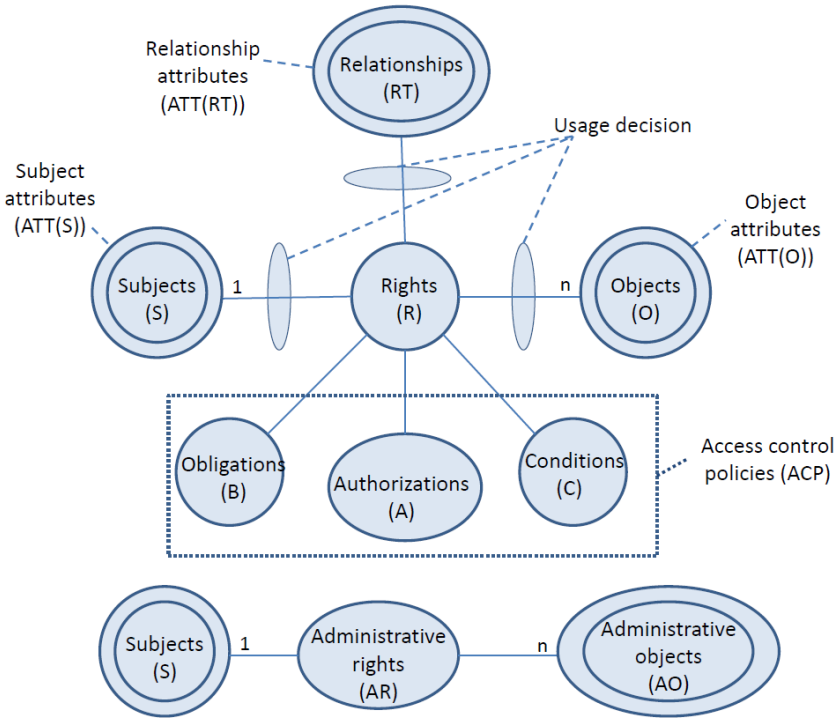


Figure 5.1: $SoNeUCON_{ADM}$

(ACP). Depicted in Figure 5.1, in $SoNeUCON_{ADM}$ owners grant use rights R over objects O regarding policies ACP and execute administrative rights AR over administrative objects AO. In this regard, following Sections describe use rights R and administrative rights AR management (Section 5.2.1 and 5.2.2 respectively).

5.2.1 Use rights management

Each owner specifies as many access control policies as desired and leaves them in a pool of policies to be evaluated when a request is received for executing some right over one of his owned objects. Contrary to other models, policies in ACP are not directly associated with data and its owner but to the owner exclusively. For instance, the policy "*grant read access to data entitled PARTY to users older than 20*" is created, associated with an owner and located in his pool of policies. Next, when an object of a particular owner is requested, all policies associated with him are evaluated, verifying authorizations (A), composed of subjects, objects and

relationship attributes and the granted right ($ATT(S)$, $ATT(O)$, $ATT(RT)$ and r), obligations (∂_b) and conditions (∂_c). If there is a policy ρ_i within the set of policies defined by an owner (P_{ow_i}) that matches the request, the right r over the requested object o is granted to the requester s. Assuming that the expression $owner(\text{“}element\text{”})$ means being owner of “element”, it is formally defined as:

$$(s, o, r)\ granted \Leftarrow P_{ow_i} = \{\rho_i \in ACP / owner(\rho_i) = owner(o)\} \wedge$$
$$\exists \rho_i (A(ATT(S), ATT(O), ATT(RT), r); \partial_b; \partial_c) \in P_{ow_i} /$$
$$\rho_i (A(ATT(s), ATT(o), ATT(rt(owner(o), s)), r); \partial_b; \partial_c) = true$$

5.2.2 Administrative rights management

This Section details the management of administrative objects (AO), revocation and delegation. In general, being owner of a particular administrative object ao grants administrative rights AR over it to manage the object and its attributes and to delegate and revoke use rights R and administrative rights AR over it. It is formally defined as:

$(s, ao, management)$ granted$\Leftarrow s = owner(ao)$

$(s, ao, delegation)$ granted$\Leftarrow s = owner(ao)$

$(s, ao, revocation)$ granted$\Leftarrow s = owner(ao)$

5.2.2.1 Administrative objects management

The management of administrative objects AO is based on the creation, the modification and the deletion of owned elements (see Figure 5.2). Specifically, as aforementioned, AO consists of S, $ATT(S)$, O, $ATT(O)$, E, $ATT(E)$ and ACP.

Concerning S, users can create an account in every WBSN which they want to be enrolled in and then, they become owners of each established access control policy, identity data and uploaded object. Likewise, WBSN users may cancel the account at any time. In terms of $ATT(S)$, the attributes of the user associated to an account can be established by its owner, retrieved from an IdP where they

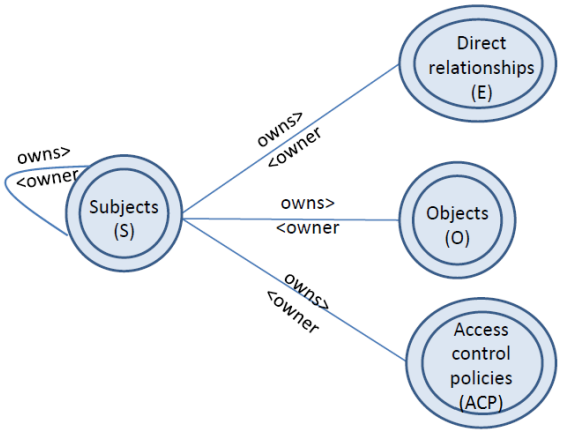


Figure 5.2: Administrative objects (AO) management

may have been previously stored or obtained from personal devices, e.g. identity cards may store the value of attribute BIRTH DATE. Indeed, these attributes are considered identity data of WBSN users.

In relation to O, owners upload objects, referred to as resources like photos, to WBSNs. In centralized WBSNs like Facebook these objects are stored in DBs owned by the WBSNs themselves. By contrast, decentralized WBSNs like Diaspora allow the storage of objects in chosen hosts. Moreover, uploaded objects should be deleted whenever desired. Regarding $ATT(O)$, the attributes of the objects owned by a user can be defined by this user, e.g. attribute TITLE, or obtained from objects metadata, e.g. LOCATION. However, if required, owners have to allow WBSNs to process objects metadata and automatically establish attributes values.

On the other hand, regarding E, users can create, update or delete direct relationships (with the edge starting at themselves and finishing at other user) and the attribute values of these relationships $ATT(E)$. Besides, $ATT(E)$ are considered identity data and then, they can be stored and thus retrieved, from IdPs.

Lastly, owners can create, update or delete access control policies ACP which include the specification of A, ∂_b and ∂_c. Recalling Section 4.3, each access control policy ρ consists of $(\rho_s; \rho_o; \rho_{rt}; r; \partial_b; \partial_c)$.

As a final remark, it is noticeable that, though $ATT(S)$, $ATT(O)$ and $ATT(E)$

are open sets of attributes, their use is bounded by WBSNs, because only those which are supported can be applied. It is the same case as with ∂_b and ∂_c involved in ACP, their use depends on available options.

5.2.2.2 Delegation management

Delegation focuses on giving permissions over an object to other users for a period of time or permanently. Delegating R can be compared with the establishment of access control control policies, if users who request a particular $r \in R$ over an object satisfy established policies, r is granted.

On the contrary, the delegation of AR requires the definition of the following function:

- DELEGATE(v_k,v_j,o_i,λ): It states that v_k gives a specific AR λ to v_j over o_i. λ refers to a partial delegation, to delegate some AR, or a complete delegation, to delegate all AR and change the owner of the delegated object. In case of complete delegation, λ takes the value $*$. Conversely, in a partial delegation, λ may take the value, e.g., AR-R to express that only the permission to grant use rights R is delegated. Note that in the administrative model proposed herein, this operation is permanent, being left for future work the management of temporal delegation.

In $SoNeUCON_{ADM}$ the delegation of AR compels the permanent delegation of all AR. Thus, the object over which the operation is executed, becomes property of the delegatee. The delegation operation should be enforced such as λ takes the value $*$, DELEGATE(v_k,v_j,o_i,$*$).

5.2.2.3 Revocation management

Revocation undoes the effect of delegation. In other words, it is the operation that undoes the granting of a right over an object to a user. As identified in Chapter 2.6, two types of revocation are distinguished, weak and strong. However,

$SoNeUCON_{ADM}$ only manages weak revocation of use rights R because recursive delegations are not applied and administrative rights AR are permanently delegated.

Specifically, in $SoNeUCON_{ADM}$ revocation is the result of the modification, either of attributes or policies. For instance, assuming that objects entitled "work" are accessible to co-workers, this policy holds until the title of objects entitled "work" changes. Indeed, it is extremely related to usage control and the application of *mutability* and *continuity* attributes. *Mutability* refers to the fact that attributes can be updated at any time. On the other hand, *continuity* refers to the enforcement of access control along the whole usage process. Both attributes are directly related to revocation because if initial conditions change along the usage process, access decisions have to be taken again [20] and they may cause the revocation of granted rights. Based on [156], revocation can be also divided between direct and indirect:

- *Direct revocation* can be enforced, at any time, by the owners of resources and identity data. Data owners may decide to revoke rights previously granted, updating or deleting an access control policy, as well as changing attributes. For instance, if the right to access a photo entitled "Classes" is granted to relationships with role "classmates", revocations can be caused by the update of the title of the photo or by the update of the role of a classmate relationship. Likewise, if the policy "Grant access to Friends to all photos" is updated to "Grant access to Friends to photos entitled Birthday", it may prevent requesters from getting requested rights in subsequent requests or while the usage process.

  In the revocation process, apart from the data owner, the Usage Reference Monitor is the entity at stake, that is, the UDF and UEF in particular (recall Section 2.5.3.1). When policies are updated or attributes are changed, the UDF is informed about that. Afterwards, it informs the occurred event to the UEF and lastly, the UEF enforces the re-evaluation of policies.

- *Indirect revocation* is caused by uncontrolled situations. Particularly, it is

performed when access control policy attributes expire or change. "Automatic" attributes updates, either subjects, objects or relationship attributes, can cause revocation of granting rights."Automatic' means that no users interactions are required. For instance, if the right to access a photo entitled "High-school" is granted to users under 18, revocations occur when requesters turn to 18 years old. Note that "automatic" updates are specially related to attributes in which time is directly or indirectly involved.

The management is equivalent to *direct revocation* except for the fact that the UDF identifies updated attributes.

5.3 $SoNeUCON_{ADM}$ high level architecture

Following the architecture presented in $SoNeUCON_{ABC}$ (Section 4.5), administrative tasks are described accordingly. Particularly, depicted in Figure 5.3, WBSN users unload resources and identity data and establish access control policies to allow certain users have rights over their data. Administrative tasks are related to the following entities:

- *Data bases (DBs)*: policies, resources and identity data DBs.
- *Management module*: this is the core component of administrative issues and it is divided in two modules.
  - *Access control policy module*: it provides users with tools to create, upload and delete access control policies.
  - *Resources and identity data module*: it provides users with tools to upload resources and manage their attributes appropriately, e.g. the specification of their title or their description. Likewise, it also allows the specification of identity data, e.g. users' addresses or users' hobbies.

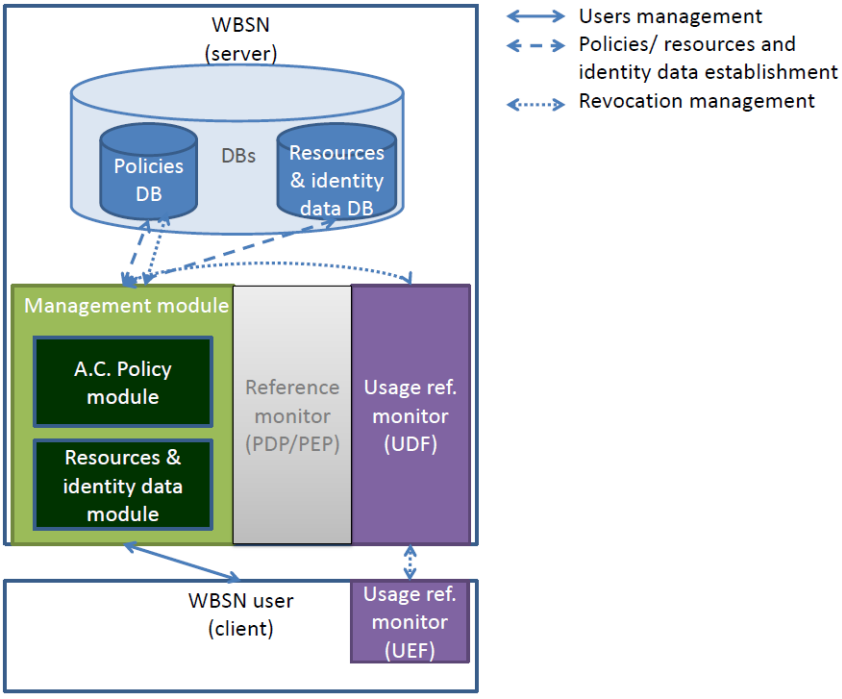


Figure 5.3: $SoNeUCON_{ADM}$ high level architecture

- *Usage reference monitor*: in case revocations take place, this component acts accordingly. The UDF is informed about attributes and access control policies updates and subsequently, it notifies the event to the UEF which enforces access control as required.

In regard to the proposed administrative model and its relation with this architecture, administrative issues are specially supported by the management module.

5.4 Summary of the chapter

$SoNeUCON_{ADM}$, an administrative model for $SoNeUCON_{ABC}$, has been presented. It manages administrative objects, as well as delegation and revocation of use and administrative rights. This Chapter concludes with the specification of administrative issues within the high level architecture presented in $SoNeUCON_{ABC}$.

Chapter 6

CooPeD: Co-owned Personal Data management

WBSN resources may belong to multiple users. Resources are uploaded by owners but they can be related to co-owners as well. For instance, users tagged in photos can be considered co-owners of the photo [33]. Therefore, resources management should support the preservation of owners and co-owners privacy.

Assorted techniques combine owners and co-owners preferences, being the voting scheme the most common one (recall Section 2.6.3). Nonetheless, contradictory preferences may compromise users privacy, i.e. the owner chooses to leave his resources public and a co-owner prefers to keep them private. The only proposal that solves this matter is the one proposed by K. Thomas *et al.* [34] where all user preferences are intersected to reach a full consensus. However, this is a restrictive technique and even being desirable the preservation of users privacy, a trade off between privacy and users demands is an essential requirement. In other words, a system may become useless if it is too restrictive.

On the other hand, current developments focus on grating or denying access to an entire piece of data without wondering about the fact that data can be, for instance, presented in a different way regarding owners and co-owners privacy preferences.

To contribute on this issue, this Chapter presents CooPeD (*Co-owned Personal Data Management*), a system that deals with co-ownership management of decomposable objects, being particularly focused on image-based data (photos and videos

without audio). It is focused on managing objects that are composed of parts, as it happens in [41]. Owners upload objects and manually or automatically assign parts to users to whom they are related to, being these users referred to as co-owners. Then, each owner and co-owners individually manages his privacy preferences. Fine-grained access control to each part is provided as the $SoNeUCON_{ABC}$ and $SoNeUCON_{ADM}$ usage control models are applied. Particularly, both models are extended herein to support co-ownership.

This Chapter is structured as follows. Section 6.1 presents an overview of the approach. A description of CooPeD is detailed in Section 6.2. Finally, Section 6.3 extends the high level architecture proposed in previous chapters to deal with co-ownership management.

6.1 CooPeD overview

CooPeD is assumed to be used in WBSNs applying the $SoNeUCON_{ABC}$ usage control model to manage access control. However, this model has to be extended to include the management of co-owned personal data. In particular, the proposed extension is applied to decomposable image-based objects, such that, $Object_j = \sum_i Object_j.Part_i + Object_j.Background$.

In $SoNeUCON_{ABC}$, each object, is managed by the user who uploads it, that is, its owner (also referred to as administrator). Nonetheless, the extension of the model should allow that each $Object_j.Part_i$ is managed by the user related to it, who should be its administrator and referred to as a co-owner of the object that contains the part. Note that $Object_j.Background$ is a fixed part of each object and under the proposed approach it is exclusively related to and managed by its owner. The extension affects both models, $SoNeUCON_{ABC}$ and $SoNeUCON_{ADM}$.

In general terms, once the owner and the co-owners have established their privacy preferences by specifying a set of access control policies, in their regard and based on each object request, access to parts and background is granted or denied.

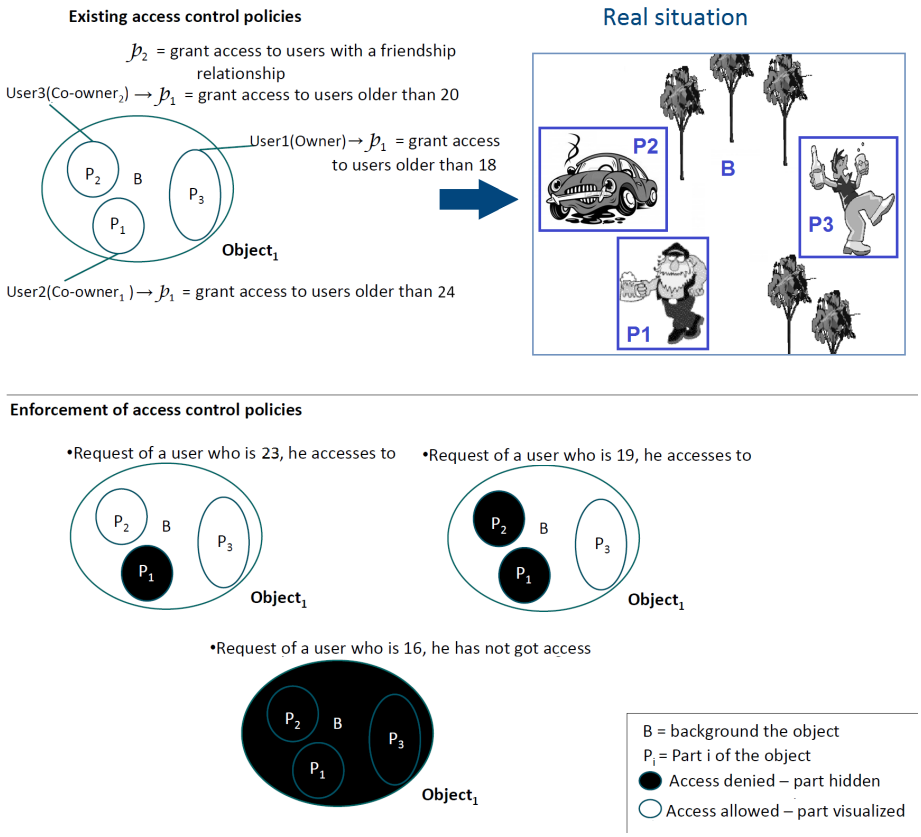


Figure 6.1: Co-ownership management of an object

If access is granted parts are visible and on the contrary, parts are hidden. An example is presented in Figure 6.1, where an object is composed of three parts in addition to the background, such that a pair of co-owners and the owner establish access control policies. The owner creates a policy to grant access to users older than 18. By contrast, co-owner$_1$ grants access to users older than 24 and co-owner$_2$ creates two policies, one to grant access to friends and other to grant access to users older than 20. Due to these restrictions three different situations are distinguished: 1) Part$_1$ is the only one hidden, e.g. a user who is 23 years old gets access to Part$_2$, Part$_3$ and the background; 2) Part$_1$ and Part$_2$ are hidden, e.g. a user who is 19 years old gets access to Part$_1$ and the background; and 3) all parts, as well as the background, are hidden, e.g. a user who is 16 years old does not get access. Therefore, access control management is privacy-preserving, considering and respecting

the privacy preferences of all users.

It is noteworthy the unnecessary development of negotiation mechanisms because each object's part is independently managed by a particular user and conflicts cannot exist.

6.2 CooPeD description

This Section presents the types of objects managed in CooPeD (Section 6.2.1) and the extension of $SoNeUCON_{ABC}$ in respect to both, access control (Section 6.2.2) and administrative management (Section 6.2.3).

6.2.1 Objects at stake

Access control looks for protecting users privacy through the protection, mainly, of users personal data. According to the EU Data Protection Directive (95/46/EC), personal data refers to "*any information relating to an identified or identifiable natural person ('Data Subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity*"[157].

As a result of above considerations and recalling the CooPeD applies to decomposable objects, parts should correspond to elements that identify or facilitate the identification of a particular user. For instance, translating into a real situation the aforementioned example (see Figure 6.1), Part_1 corresponds to the image of co-owner_1 himself, Part_2 to the image of co-owner_2's car and Part_3 to the image of the owner himself. Then, parts can be users or elements they possess, such as animals, vehicles, houses, elements they are carrying, etc. Nonetheless, objects decomposition is quite restrictive as it depends on the amount of parts that users can distinguish and automatic tools can perform. Besides, it is assumed that each object's part is assigned to a single user and the management of parts related to

multiple users is left for future work. For instance, an image of a couple of users in front of their house opens up the discussion about who has to manage the part of the image related to the house.

Therefore, this proposal is focused on a pair of image-based types of objects, photos and videos (without audio):

- *Photos* are composed of assorted elements such as users, vehicles, animals, etc., being desirable that all users related somehow to them manage access control. For this purpose, considering that H. Lipford *et al.* mentioned the appropriate use and possible application of graphical techniques to manage access control [158], the analysis of recognition techniques to identify elements within photos is essential.

- *Videos (without audio)* are compared with a sequence of photos, thereby they can be also decomposed. Nonetheless, as there are a great amount of photos per video, their management is significantly more tedious and complex.

Lastly, it is noticeable that CooPeD is based on image-based data, although other objects like documents, music or videos with audio can be also applied. Once access control policies of the owner and co-owners are verified, the requested document, music or video with audio can be processed accordingly. In the case of documents, the appropriate sentences can be hidden; in relation to music, the right notes can be silenced; and in regard to videos with audio, the appropriate tracks can be omitted. Nonetheless, managing this type of objects involves a great deal of technical complexity. For instance, in a video record (with audio), a user may say "last night I went out with Charles", being indispensable the appropriate identification of co-owners in all photo sequences as well as in the audio track until doing a complete decomposition. Therefore, this contribution focuses on image-based data and leaves for future work the management of other kinds of data.

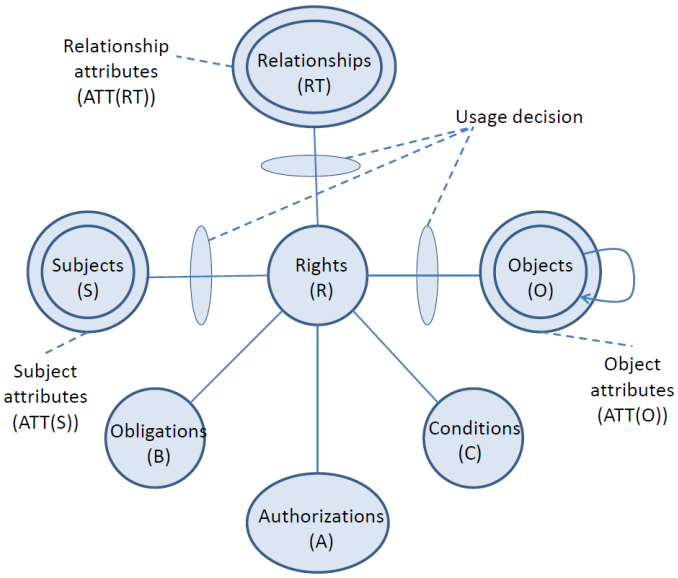


Figure 6.2: SoNeUCON coownership management

6.2.2 Extension of $SoNeUCON_{ABC}$ usage control model

$SoNeUCON_{ABC}$ is extended in such a way that the entity Objects (O) includes an additional link to state that objects are composed of objects (Figure 6.2). Each object o_i is decomposed in n objects o_i^j such that $o_i = \sum_{j=0}^{n} o_i^j$. Therefore, a tree structure $T(o_i)$, where o_i^j are the leaves, can be identified (Figure 6.2). In CooPeD, managed objects are image-based, photos and videos (without audio) which are decomposed in the background and multiple parts, that is, objects composed of objects. In this regard, a decomposed object consists of the background of the image o_i^0, and as many objects o_i^j as required. Depicted in Figure 6.3, o_1 consists of three objects being o_1^0 the background. It must be noted that decompositions can be recursively performed creating a tree $T(o_i, h)$ of depth h where each o_i^j at a given depth may become the root of a sub-tree. However, for the sake of simplicity and without losing generality only one level of decomposition is considered herein and recursion is left for future work.

Concerning attributes, the existence of o_i^j leads to the emergence of additional in $ATT(O)$. A new $att(o)$ can be the type of the part of the decomposed object.

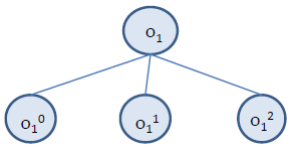


Figure 6.3: $T(o_1)$ example

Then, analogously to $SoNeUCON_{ABC}$, $ATT(O)$ can be derived from the mapping $dAT : O \longrightarrow ATT(O)$. Furthermore, given that an o_i is composed of several o_i^j, each o_i^j inherits at first $att(o_i)$ from its parent object.

6.2.2.1 Access control policies specification

In terms of access control policies, their structure remains as $\rho(\rho_s; \rho_o; \rho_{rt}; r; \partial_b; \partial_c)$ (see Chapter 4). The main change is that ρ_o can involve additional $ATT(O)$. Indeed, the use of these attributes helps to reach fine-grained access control policies particularly when co-ownership management takes place. The $att(o)$ $partType$ would help to determine the precise type of an object o_i^j to which the policy applies. If $partType$ takes the value $user$, it means that the o_i^j to whom it is attached refers to the image of the owner/ co-owner himself. Similarly, if $partType$ takes value car, it states that the o_i^j to whom it is attached corresponds to the owner's/ co-owner's car.

In Figure 6.1, the proposed example depicts an object o_i which is decomposed in four parts $\{B, P_1, P_2, P_3\}$ where B refers to the background. In the example the established access control policies are pointed out below. All users specify $partType$ to reach fine-granularity and guarantee that their established policies are enforced when o_i^j come into play:

- User1 (Owner:)

  $\rho_1 = ((age > 18);\ \varnothing;\ partType = User;\ \varnothing; Read;\ \varnothing;\ \varnothing)$

- User2 (Co-owner$_1$):

  $\rho_1 = ((age > 24);\ \varnothing;\ partType = User;\ \varnothing; Read;\ \varnothing;\ \varnothing)$

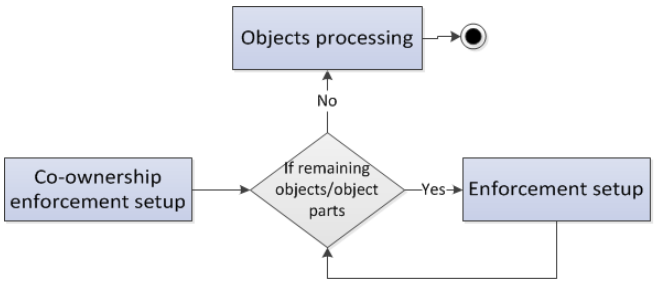


Figure 6.4: Activity diagram of the enforcement process.

- User3 (Co-owner$_2$):

  ρ_1 = $(\varnothing;\ \varnothing;\ partType = car;\ ((((role = friend))), \varnothing,\ \varnothing);\ Read;\ \varnothing;\ \varnothing)$

  ρ_2 = $((age > 20);\ \varnothing;\ partType = car;\ \varnothing; Read;\ \varnothing;\ \varnothing)$

6.2.2.2 Access control policies enforcement

When $r \in R$ over $o_i \in O$ is requested and o_i has been decomposed in o_i^j objects, co-ownership management starts. Access control enforcement is divided in three main activities, see Figure 6.4. First, co-ownership enforcement setup is carried out, identifying involved co-owners and parts o_i^j of the requested object. Secondly, according to each o_i^j of the owner and of the co-owners, the enforcement of their access control policies is performed as in $SoNeUCON_{ABC}$ (Section 4.4). Finally, after the evaluation of the owner and the co-owners policies, the requested object o_i is processed accordingly.

Again, as in $SoNeUCON_{ABC}$, the work-flow of the enforcement process in terms of applied activities and actions involved in them is depicted in Figure 6.5. Likewise, actions correspond to developed functions and they are pointed out in square brackets. The enforcement starts identifying parts o_i^j of the requested object o_i that belong to the owner [$FindObjects$]. Similarly, co-owners linked to the requested object o_i are noticed [$FindCoOwners$] and objects parts o_i^j attached to each of them are identified [$FindObjects$].

Concerning the activity of enforcement setup, the verification of access control

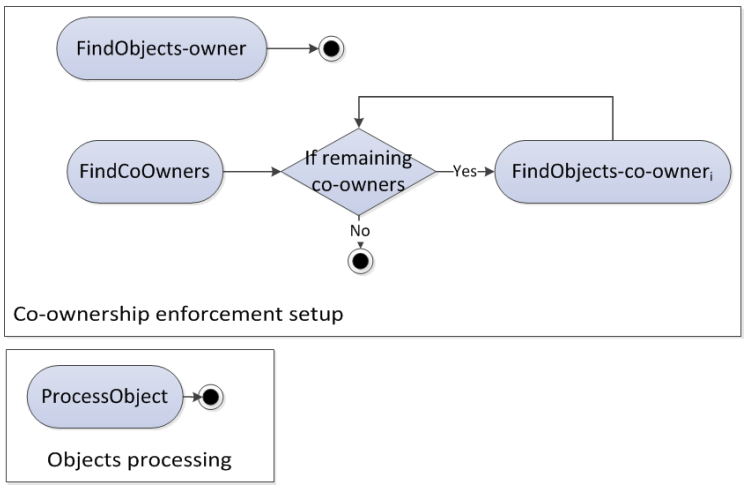


Figure 6.5: Enforcement process work-flow.

policies is carried out analogously to $SoNeUCON_{ABC}$ (recall Section 4.4).

Finally, objects are processed according to the owner and the co-owners access control policies, and the requested right r is (or is not) granted [$ProcessObject$]. In particular, if the request matches the conditions of an access control policy of the owner and an access control policy of each co-owner, the requested right r over the requested object o_i is granted. By contrast, if the request does not match conditions of any policy of the owner and any policy of the co-owners, r is not granted. On the other hand, if the request matches conditions of a policy of the owner or with the conditions of a policy of some co-owners, o_i is processed and r over appropriate o_i^j is granted.

Note that the concrete specification of every function is presented in Appendix D.2.

6.2.3 Extension of $SoNeUCON_{ADM}$

First of all, the different between R and AR should be recalled (Chapter 5). R refer to operations that can be performed with objects, e.g. read, move, copy, etc. On the contrary, AR correspond to the management of elements involve in the access control decision process, as well as delegation and revocation.

In the extension of $SoNeUCON_{ADM}$, described in the following Sections, AR management is extended to include decomposition management (Section 6.2.3.1), as well as changes in terms of delegation (Section 6.2.3.2). The rest of administrative functions, namely, the management of use rights R, administrative objects AO and revocation, remain as in $SoNeUCON_{ADM}$.

Furthermore, a total of three issues should be noted. Firstly, the background of an image is always managed by owners. Even though no other o_i^j would be attached to the owner, he is in the possession of the background to delegate R over it. Secondly, in case a given o_i^j could not be assigned to any user, e.g. the related user is not a WBSN user, it would be attached and managed by the owner as well. Lastly, it should be pointed out that administrative objects (AO) involve decomposable objects and then, for an object o each o_i^j is also considered an ao.

6.2.3.1 Decomposition management

Each object o_i is uploaded by a user, the owner, who owns R and AR over it. Then, if o_i can be decomposed in o_i^j, the owner (or the WBSN on his behalf) decomposes it and assigns a co-owner to each o_i^j. As a result, co-owners owns use rights R and administrative rights AR over their o_i^j. Decomposition is formally defined as:

$(s, ao, decomposition)$ granted$\Leftarrow s = owner(ao) \wedge ao \in O \wedge ao = \{set\ of\ o_i^j\}$

6.2.3.2 Delegation management

Also recalling Chapter 5, *delegation* focuses on giving permissions to other users over certain elements. Delegating R can be compared with the establishment of access control control policies. By contrast, the delegation of AR, which is assumed to be permanent, applies the function DELEGATE(v_k,v_j,o,λ) where λ takes the value $*$. Besides, it can be executed manually or automatically. The owner of an o_i can manually identify o_i^j and delegate administrative rights AR accordingly. By contrast, a WBSN can automatically detect users linked to o_i^j and enforce the

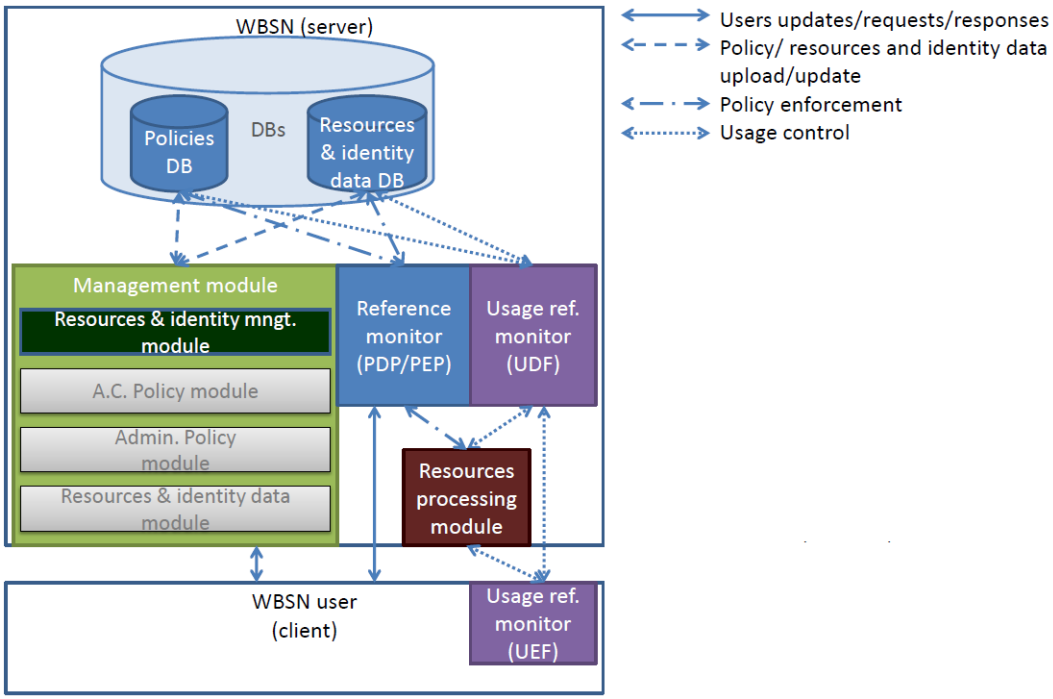


Figure 6.6: Co-ownership management high level architecture

delegation on the owner's behalf. Moreover, both the WBSN and the owner are assumed to be trusted to execute this operation.

Note that, from a privacy point of view, the permanent delegation of AR is an essential requirement because it is assumed that each o_i^j should be always managed by its owner/ co-owner. Then, though the operation DELEGATE may consider temporal delegations in the future work, co-ownership management should preferably apply permanent delegations.

6.3 Co-ownership high level architecture

Co-ownership management requires the development and deployment of the architecture depicted in Figure 6.6. It is rather similar to the one proposed in $SoNeUCON_{ABC}$ except for the inclusion of some new tasks and a new pair of modules:

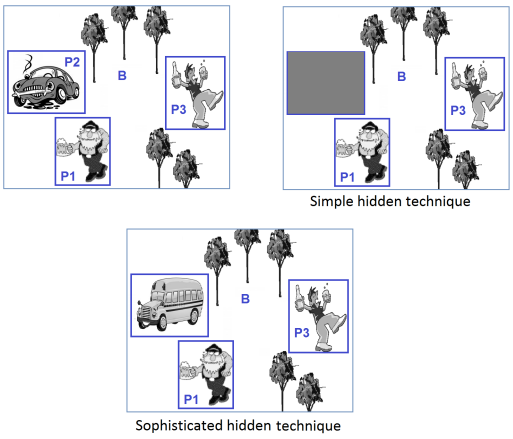


Figure 6.7: Hidden techniques

- *Data bases (DBs)*: there are DB for policies, resources and identity data. DBs have to store the resources and attach to them their decomposed parts and the identifiers of co-owners. Recall that objects are analogous to resources.

- *Management module*: this module is in charge of administrative operations. Specifically, the module *Resources management module* is introduced to provide tools to decompose resources and link each of them to the appropriate co-owners.

- *Reference monitor*: it is responsible for the enforcement process. Given users' requests, it verifies access control policies of owners and co-owners and grants or denies rights accordingly. Moreover, before granting requested rights over requested resources, it interacts with the *Resources processing module* to process the policies related to those resources.

- *Usage reference monitor*: it enforces access control at usage time. The main issue related to co-ownership management is based on the need of identifying changes in owners access control policies and changes in co-owners access control policies. The UDF has to identify occurred events and the UEF enforces access control. Nonetheless, some changes may require processing used resources and thus, interactions with the *Resources processing module* are

required.

- *Resources processing module*: this is a new module introduced to manage co-ownership. Specifically, it provides tools to hide parts of resources according to the enforcement of access control policies. Note that hiding techniques can be very assorted, see Figure 6.7. A simple technique may focus on covering resources with opaque figures, while a more sophisticated one may focus on replacing a resource part (of a decomposed resource) with another that prevents the identification of the replaced part.

6.4 Summary of the chapter

WBSNs manage resources of multiple users and some of them are not related to a single owner, but to multiple co-owners as well. To solve privacy problems caused by co-ownership management this Chapter has presented CooPeD, a mechanism to protect all users privacy through the management of decomposable objects. It is developed over $SoNeUCON_{ABC}$ and $SoNeUCON_{ADM}$ usage control models. Specifically, access control management and administrative management are described in regard to previous models, together with the applied high level architecture.

Chapter 7

UMA+FOAF Social Network Protocol. Achieving interoperability and reusability between WBSNs

---

Assorted services are offered by different WBSNs to encourage their involvement. To reduce the burden of creating multiple accounts, uploading data and establishing access control policies, this contribution proposes a protocol towards interoperability and reusability.

In general, developments focused on interoperability may reach resuability as well, though there are some exceptions. For instance, [101] addresses interoperability of policies but as they have to be stored in each WBSN, reusability is not achieved. However, recalling Chapter 3, though interoperability and reusability have been independently noticed, their joint application has not been addressed.

In WBSNs resources, identity data and access control policies are the data at stake (recall Section 2.1). Some approaches look for providing interoperability either of resources or identity data or access control policies and just [1] provides interoperable resources and policies. Nonetheless, neither interoperability nor reusability is managed in respect to identity data, resources and access control policies simultaneously.

This Chapter presents an architecture and the complementary protocol to attain interoperability and reusability between users directly connected who are enrolled in different WBSNs. Note that this protocol allows considering contacts enrolled in different WBSNs, thus Alice in LinkedIn may have Bob of Facebook as a contact. The proposed solution is based on decentralizing access control policies, resources and identity data management. The main contribution is the development of the UMA+FOAF Social Network Protocol (U+F). This protocol is based on the User-Managed Access (UMA) core protocol [1] to address the decentralization of resources, identity data and access control policies, and on the Friend-Of-A-Friend (FOAF) project [42] to address the description of identity data. The $SoNeUCON_{ABC}$ model is taken as the underlying base to manage access control (recall Chapter 4).

First, an overview of the approach is presented in Section 7.1. Afterwards, the system model is proposed in Section 7.2, describing the proposed architecture, the requirements, the trust and adversarial model, and the applied FOAF files. Finally, the description of the protocol is detailed in Section 7.3.

7.1 System overview

Currently, each WBSN provides the storage of personal data, access control policies and resources and the establishment of relationships, resource uploads and updates. WBSNs can be depicted as different worlds. For instance, assuming that a user has a Facebook account and other user has a MySpace account and each of them wants to visualize resources and identity data of the other, these activities are infeasible.

In order to simplify access control management in WBSNs, based on a particular application of UMA and FOAF, UMA+FOAF Social Network Protocol (U+F) is presented.

Described in Section 3.4.3, UMA is applied to manage data no matter where they live on the web. In the social networking context, WBSN users, referred to as

Authorizing Users (AUs) in UMA, establish resources in chosen Hosts, identity data in chosen IdPs and access control policies in chosen Authorization Managers (AMs). Moreover, given the amount of entities that may exist in the system proposed herein, Certification Authorities (CA) provide certificates to attest entities' trustworthiness. Then, once data is located in chosen repositories, WBSNs, on behalf of users, act as UMA Requesting Parties (RPs). Per each resource or identity data request, WBSNs contact with the right Host or IdP which redirects them to the appropriate AM to verify access control policies. However, access control policies verification requires the satisfaction of claims, which refer to structures with all necessary data for the policies verification process. Given that the U+F proposal considers direct relationships, claims should refer to, at least, an accreditation of the fact that the requester has some kind of relationship with the owner of the requested resource or identity data. In U+F, this accreditation is obtained from the IdP of the owner of the requested resource or identity. Afterwards, when WBSNs provide AMs the requested claims, if access control policies are satisfied, the AMs deliver access tokens. Finally, access is granted when tokens are presented to the right Host or IdP. As a result of this description, depicted in Figure 7.1, U+F allows that a pair of users, User1 and User2, even being enrolled in Facebook and MySpace respectively, may interact with each other because identity data and resources are accessed through both WBSNs. Moreover, given that User1 has other account in Badoo, if desired, resources, access control policies and identity data used in Facebook can also be used in this WBSN, reaching reusability. Note that in this example, an IdP, a Host and a pair of AMs are applied per user. Nevertheless, each user can choose the amount of IdPs, Hosts and AMs to use, e.g. a Host may offer limited storage per user and some of them may be required.

Furthermore, identity data, composed of users profiles and contacts data, can be described in different ways. In relation to this, given the flexibility FOAF provides (recall Section 3.5), identity data is described using FOAF.

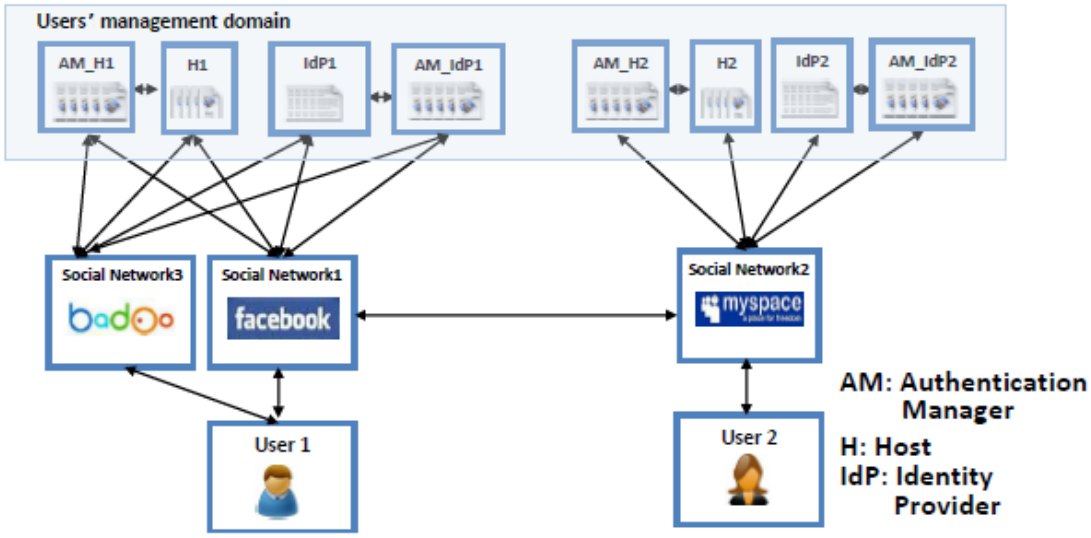


Figure 7.1: WBSN applying U+F

In further detail, U+F is composed of three phases. The first phase is the **initialization**. It refers to the configuration of entities and elements involved in the protocol. Subsequently, the second phase starts when a **user logins in a WBSN**. At this moment, the user accesses to his identity data and contacts data, which are stored in the chosen IdP. Besides, his resources, stored in a particular Host, remain available. The last phase is the **access data of a direct contact** who is enrolled in a different WBSN. In particular, this last phase is divided in accessing to the contact's identity data and accessing to the contact's resources.

Last but not least, access control is based on $SoNeUCON_{ABC}$. WBSN users are the subjects (S) who own resources, referred to as objects (O), with attributes attached to them ($ATT(O)$). Additionally, users own profiles composed of values of users attributes ($ATT(S)$) and their contacts' data, referred to as direct relationships (E) with attributes attached ($ATT(E)$). Nonetheless, in this Chapter two simplifications on the $SoNeUCON_{ABC}$ model are assumed. The first simplification is that only direct relationships between WBSN users are considered (that is a user specifies having a direct relationship with other users to express that they are his contacts). However, access control management in U+F assumes the existence of

a bidirectional relationship between the requester and the administrator of a requested resource or identity data. Then, for instance, if User_A wants to access to a resource of User_B, it is required that User_A has specified having a relationship with User_B, as well as User_B has specified having a relationship with User_A. These direct relationships must be stored in each user's FOAF file. The second simplification considered herein is that access control policies exclusively involve $ATT(S)$, $ATT(O)$ and $ATT(E)$. Thus, obligations (B) and conditions (C) are left out of the protocol's scope.

Furthermore, regarding access control enforcement, U+F also focuses on a simplified version of $SoNeUCON_{ABC}$. In this usage control model, access control enforcement starts by constructing the rt structure, composed of all enriched paths between the requester and the administrator of a requested object, and continuous with the evaluation of access control policies in regard to rt (recall Section 4.4). By contrast, in U+F, rt is not constructed. The use of rt is specially useful for indirect relationships management and as U+F manages direct relationships, the construction of rt is discussed in the extension of U+F presented in Chapter 8. Likewise, the consideration of usage control and all identified WBSN features, namely, *direction*, *distance*, *multi-path*, *common-contact*, *clique* and *flexible elements in access control policies*, are also discussed in Chapter 8.

7.2 System model

U+F model involves the definition of the architecture (Section 7.2.1), the system requirements (Section 7.2.2), the applied FOAF files (Section 7.2.3) and the trust and adversarial models (Sections 7.2.4 and 7.2.5 respectively).

7.2.1 Architecture

U+F is composed of an architecture and the associated protocol. In regard to the architecture, it is composed of the five entities described below (Figure 7.2). Note

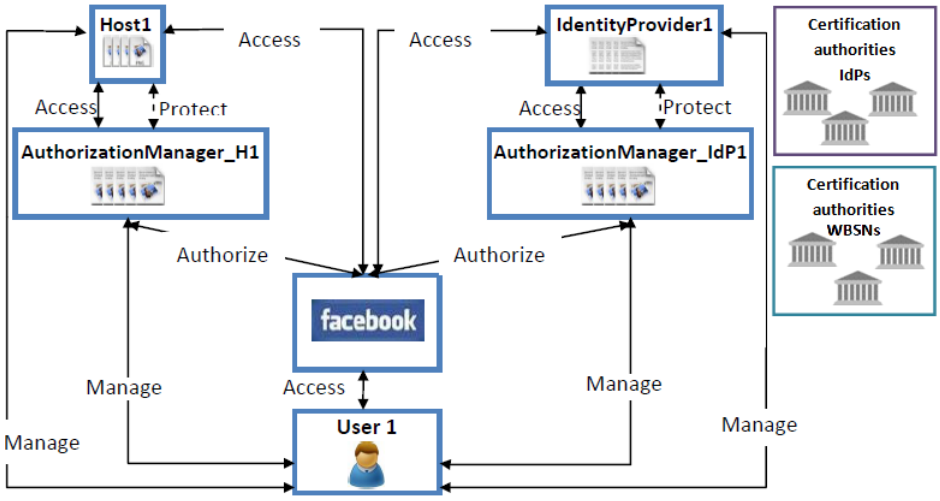


Figure 7.2: U+F architecture

that, as mentioned above, multiple Hosts, IdPs and AMs can be used but, for the sake of clarity, a single Host and a single IdP per user and an AM per each of these entities are considered.

1. *User* (U): a user has different roles. On the one hand, a user plays the role of a UMA's Requesting Party (RP) who is able to access resources of his contacts through WBSNs. On the other hand, a user also plays the role of an Authorizing User (AU) by performing three main operations, (1) placement of resources in his Host together with later updates of them, (2) deployment of his FOAF file in his Identity Provider (IdP) and (3) deployment of policies in his Authorization Managers in respect to resources and identity data.

2. *Identity provider* (IdP): repository of FOAF files which are placed by AUs, as well as provider of claims. Furthermore, IdPs act as a PEP when providing FOAF files. Notice that this entity is a Host but instead of storing resources, it stores identity data. Besides, to manage claims, per each user, IdPs store a list of IdP Certification Authorities (IdP_CAs) that each user considers reliable. Moreover, to guarantee communications with WBSNs that are trusted by users, per user, a list of WBSN Certification Authorities (WBSN_CAs) which

are considered trustworthy is also stored.

One last remark is that IdPs can refer to personal servers of WBSNs users or servers of particular companies.

3. *Host*: repository of resources, analogous to a data base service, in which the AU stores resources. Moreover, it acts as a PEP, delivering resources once the right token is presented.

   Analogous to IdPs, Hosts may refer to personal servers of WBSNs users or servers of particular companies.

4. *Authorization Manager* (AM): entity that evaluates policies previously established by an AU. However, to achieve this purpose the AM requests claims to perform policy validation and delivers tokens. Therefore, each AM acts as a PDP and plays the conceptual role of a Security Token Service. Also, in order to verify claims, they store, per each user, a list of the IdP_CAs trusted by the user. Likewise, to communicate with WBSNs considered trusted by users, per user, a list of WBSN_CAs which are considered reliable is also stored. Note that these lists have to be analogous (for consistency) to the ones established in IdPs and then, AMs can contact IdPs to obtain both lists before the protocol's execution.

   On the other hand, concerning access control policies, they are defined over $SoNeUCON_{ABC}$, namely using objects, subjects and direct relationship attributes to some extent ($ATT(S)$, $ATT(O)$ and $ATT(E)$).

5. *Social Networks*: referred to as WBSNs, provide an interface to show resources and identity data and also, provide the management of wall comments, resource comments and any other extra services. Moreover, this entity acquires the role of the requester in UMA and performs three main operations: (1) it acts on behalf of a RP and interacts with Hosts to reach protected resources; (2) it interacts with AMs to get the appropriate token in regard to requested

resources; and (3) it interacts with the adequate IdP to get users' personal data each time a user session starts. This last operation is performed after the authentication between the user and his Host and IdP.

Each WBSN owns a certificate generated by a WBSN Certification Authority (WBSN_CA). Therefore, once a request is sent from a WBSN to another, taking jointly the role of a requester, called herein Fat Requester, certificates authenticate both WBSNs. Note that this is one difference concerning UMA, that is, a pair of entities act as a single one.

6. *Certification authorities* (CA): these entities are in charge of delivering certificates to trusted entities to allow them signing interchanged messages. They can be compared with traditional certification authorities that issue public key certificates. The delivery of certificates is carried out according to particular criteria and rules whose specification is left as an open research issue.

   A pair of groups are distinguished. A first group provides certificates to IdPs (IdP_CAs) and another group to WBSNs (WBSN_CAs). Then, per user, AMs and IdPs store a list of IdP_CAs to ensure, along the protocol execution, that claims are provided from trusted IdPs. Likewise, IdPs and AMs store, per user, a list of WBSN_CAs to ensure that interoperability is only allowed between trusted WBSNs. Thus, for instance, a User_A enrolled in a WBSN_A cannot access resources or identity data of a User_B enrolled in the WBSN_B, unless User_A trusts the WBSN_CA that has issued the certificate to WBSN_B, that is, that WBSN_CA is listed in his WBSN_CA list stored in AMs and IdPs.

   The existence of groups of CAs instead of a single entity considered the root is due to the worldwide use of these applications. Huge quantity of people in multiple countries make use of WBSNs and a central entity would be impractical. Likewise, also looking for the simplification of the management of certificates, there are CAs to independently certify IdPs and WBSNs.

7.2.2 Requirements

According to U+F features, the following requirements have to be achieved:

1. **Interoperability and reusability in regard to direct relationships.** The communication and interchange of data between multiple users enrolled in different WBSNs has to be attained.

2. **Resources and identity data confidentiality and access control.** Resources and identity data have to be exclusively delivered and used by authorized users and entities involved in the protocol.

3. **Resources and identity data integrity.** Resources and identity data do not have to be altered along the protocol execution.

4. **Chain of trust.** Given the great set of entities at stake, the final receiver has to be able to verify that entities through which interchanged messages pass are trusted.

5. **Access minimum identity data**. The amount of identity data accessible by WBSNs has to be minimized. Once a WBSN accesses to a user's identity data, the management has to be carried out using the least possible data. Indeed, this is directly related to "The principle of least privilege" which is based on the fact that every program should operate using the least possible amount of privileges [159]. In particular, this is called data minimization [160] and it can be identified as a common principle in the development of Privacy Enhancing Technologies.

7.2.3 Personal file structure

Identity data refers to the own user profile and the user's contacts data which are structured in a FOAF file specially developed for U+F. FOAF is particularly used to describe people and their contacts relationships. In the FOAF project specification

[146] several attributes are already established, like "familyName", "homepage" or "interest", related to a particular user, and "knows", that refers to the existence of a relationship with someone. However, apart from these possibilities offered by FOAF, some other attributes have been created in this work (see Figure 7.3).

Regarding profile data, the attribute "nationality" is proposed. It refers to the nation where the user comes from and "WBSNs", attribute which refers to the name of WBSNs, separated by a space, that the contact is registered in. Moreover, profile data consists of, at least, the user name and the user email address that, for security reasons, is stored after having applied a hash function to it.

On the other hand, in what concerns relationship data, the following attributes per direct relationships have been developed: "creation date", that refers to the date when the relationship was created; "trust", that corresponds to a low, medium or high trust level related to the relationship; "duration", which corresponds to the period in which the relationship remains valid and it includes a starting and an ending date; and "WBSNs" analogous to the one mentioned above. Moreover, regarding relationships, they are unidirectional and supposing that a user, called Bob, has a work relationship with a user called Alice, his FOAF file includes Alice's relationship but it does not necessarily in the other way round. The FOAF file depicted in Figure 7.3 corresponds to this example.

Nonetheless, in the protocol, reduced FOAF files are also used, called in [61] sub-profiles. In general terms, they correspond to FOAF files without relationships information. In U+F, these files are used in the acquisition of claims. Furthermore, the quantity of included user attributes depends on access control policies. For instance, some users may have access to nationality while others may not.

7.2.4 Trust model

In U+F the trust model is based on the following assumptions:

```
<rdf:RDF>
    ...
    <foaf:Person rdf:nodeID="RequesterBob">
        <foaf:name>Bob Romero Gutierrez</foaf:name>
        <foaf:mbox
        sha1sum>8567c8b121_cd99604a40jhf552a2d884c234b3</foaf:mbox
        sha1sum>
        <att:nationality>Spanish</att:nationality>
        <att:WBSNs>facebook badoo</att:WBSNs>
        <relation:worksWith rdf:resource="#RequesterAlice">
            <relation:trustRel>highTrust</relation:trustRel>
            <relation:relCreationDate>2011-10-03</relation:relCreationDate>
            <relation:relDuration>
                <relation:relDurationIni>2011-10-03</relation:relDurationIni>
                <relation:relDurationFin>2013-10-03</relation:relDurationFin>
            </relation:relDuration>
        </relation:worksWith>
    </foaf:Person>
        <foaf:Person rdf:nodeID="RequesterAlice">
        <foaf:name>Alice Gutierrez Gamella</foaf:name>
        <foaf:mbox
        sha1sum>8567c8b121_cd99604a40jhf552a2d884c234b3</foaf:mbox
        sha1sum>
        <foaf:WBSNs>myspace picasa</foaf:WBSNs>
    </foaf:Person>
</rdf:RDF>
```

Figure 7.3: Proposed FOAF file including new fields

- IdPs, Hosts and AMs are trusted in the sense that they provide secure storage and they are honest to guarantee no protocol deviation.
- WBSNs provide secure data management by not disclosing users resources or identity data. However, they may attempt to impersonate users.
- Trust management is left to CAs that issue certificates to trusted IdPs and WBSNs according to a set of criteria not defined in this work.

7.2.5 Adversarial model

An adversary in this model is an entity that can perform the following actions:

- It can inject and alter data to provide unauthorized accesses to resources or identity data.
- It can can get tokens some time after being delivered by AMs to get access to resources or identity data. Besides, an adversary can access any other data interchanged in the protocol.
- It may try impersonating IdPs and WBSNs to, again, get access to resources or identity data.

Note that Denial of Service (DoS) attacks can not be performed by an adversary as the protocol works on the assumption that existing techniques, e.g. for load balancing, are applied.

7.3 U+F protocol description

The description of U+F is divided in the definition of messages content (Section 7.3.1) and the description of the execution procedure (Section 7.3.2).

7.3.1 Messages content

Along the protocol an assorted set of messages is interchanged. More specifically, messages content is classified under three different categories, operations, elements and structures, where structures correspond to sets of elements over which operations are performed.

7.3.1.1 Operations

Operations involved in U+F correspond to digital signatures that are performed by WBSNs and IdPs to guarantee their trustworthiness.

7.3.1.2 Elements

In general, there are six elements within interchanged messages: *user identifiers*, *tokens*, *file identifiers*, *tickets*, *signatures* and *redirections*.

1. All messages have *user identifiers*, which correspond to the identifiers of the requester and the administrator of a requested file. As in a great amount of applications, *user identifiers* are defined as emails though, for security reasons, they are not stored in plain text but as a hash.

2. *Tokens* should contain an expiration time. Note that, according to the OAuth specification [144], the protocol which lays the bases of UMA, tokens generally

correspond to a random string associated to a resource and an expiration time. Point out in [144], the misuse of access tokens to impersonate a resource owner is out of the scope of OAuth specification. In order to face up to impersonations attacks, in U+F WBSNs are assumed to be trusted.

By contrast, although in this protocol impersonations are avoided, on the assumption that they may exist, a possible solution could be including the identification of the administrator and the requester as part of tokens. Besides, requesters would have to sign tokens once delivered to IdPs or Hosts. Nonetheless, this alternative requires a deep analysis because it would increase the protocol's workload due to the new set of required operations and interchanged messages.

3. *File identifiers*, as its name suggests, each of them identifies a particular WBSN file, either identity data (a FOAF file) or a resource.

4. Once a particular piece of data is requested to an IdP or Host, it delivers a *ticket* that identifies the requested data and it is used to acquire the access token in the right AM.

5. Each time a *signature* is performed the serial number of the public key certificate of the signing entity and the signature date and time is attached to it. Notice that some messages interchanged along the protocol are signed and, to do it properly, signing entities have to acquire the time from a trusted site such as the NIST Internet Time Service[1]. Besides, signatures have to be identified as short-term certificates issued by trusted entities (note that a trusted entity depends on users' preferences). Such signatures are specially significant in messages interchanged to acquire claims because they certify certain users attributes and a particular relationship between a pair of users.

6. *Redirections* refer to the location of the entities to which redirections are

[1] http://www.nist.gov/pml/div688/grp40/its.cfm , last access May 2014

Table 7.1: Interchanged messages in U+F

Id	Name	Content
M1	Token request	Ticket $\|WBSN_R\_Cert\_Serial\_Number\|$ Date_time$_{WBSN_R signature}$|S$_{k_{WBSN_R\text{-}Cert}}$(Complete Message)
M2	Token request redirection	Ticket $\|AM\_location$
M3	Token response redirection	Ticket $\|Tokenvalue$
M4	Token response	Ticket $\|Tokenvalue\|WBSN_R\_Cert\_Serial\_Number\|$ Date_time$_{WBSN_R signature}$|S$_{k_{WBSN_R\text{-}Cert}}$(Complete Message)
M5	File request	R_Id $\|File\_Id$
M6	File indirect request	R_Id $\|A\_Id\|File\_Id\|WBSN_R\_Cert\_Serial\_Number\|$ Date_time$_{WBSN_R signature}$|S$_{k_{WBSN_R\text{-}Cert}}$(Complete Message)
M7	File response	R_Id $\|File$
M8	Claims request	R_Id $\|A\_Id\|DataRrequest$
M9	Claims structures response	R_Id $\|A\_Id\|AccreditationR\|DataRresponse\|$ IdP$_R_Cert_Serial_Number\|Date_time_{IdP_R signature}\|$ $S_{k_{IdP_R}}$(Complete Message)
M10	Claims response	Claims structures response $\|RelationshipR\_A\|$ IdP$_A_Cert_Serial_Number\|Date_time_{IdP_A signature}\|$ $S_{k_{IdP_A}}$(Relationship R_A) $\|WBSN_R\_Cert\_Serial\_Number\|$
M11	Certify direct relationship	R_Id $\|A\_Id\|AccreditationR\|IdP_R\_Cert\_Serial\_Number\|$ Date_time$_{IdP_R signature}$|S$_{k_{IdP_R}}$(Accreditation R)| Relationship R_A $\|WBSN_R\_Cert\_Serial\_Number\|$ Date_time$_{WBSN_R signature}$|S$_{k_{WBSN_R\text{-}Cert}}$(Complete message)
M12	Relationship certified	R_Id $\|A\_Id\|RelationshipR\_A\|IdP_A\_Cert\_Serial\_Number\|$ Date_time$_{IdP_A signature}$|S$_{k_{IdP_A}}$(Relationship R_A)
M13	Simple token request	Ticket
M14	Simple token response	Ticket $\|Tokenvalue$
M15	Token validation	Ticket $\|Tokenvalue$
M16	Simple claim request	R_Id $\|AccreditationRrequest$
M17	Simple claim response	R_Id $\|IdP_R\_Cert\_Serial\_Number\|AccreditationR\|$ Date_time$_{IdP_R signature}$|S$_{k_{IdP_R}}$(Accreditation R)

performed and, in particular, they are urls.

In sum, a total of 17 different messages are interchanged along the protocol. All messages and their content are described in Table 7.1, where symbol $\|$ implies concatenation and S means signature. Interchanged messages mainly follow UMA's core protocol specification [38], although some new fields haven been added in some cases and a few new messages have been specified.

7.3.1.3 Structures

In U+F there are four main types of applied structures. First, the *Accreditation* which identifies who is the requester of a particular requested file.

The second structure applied in this protocol is called *RelationshipA-B* and it refers to the identifiers of the users involved in the relationship. Moreover, this structure is particularly defined as *first: hash_email_Requester* and *end: hash_email_Administrator*.

On the other hand, a pair of structures are specially used to verify the satisfaction of each established access control policy. The first structure corresponds to *Data request* which consists of the name of the set of attributes that are involved in the applied access control policy and must be provided by the requester. It is defined as *attributes: att1 att2 att3 ....* Finally, the last structure corresponds to *Data response* which includes the values of all requested attributes in a *Data request* and it is defined as *attributes: att1 att2 att3 ... attributesData: valueAtt1 valueAtt2 valueAtt3*.

7.3.2 Execution procedure

The U+F protocol is divided in three phases: (1) the **initialization** phase, in which the user, in the role of AU, initializes all entities involved (except for CAs and other users) and, acting as RP, provides his WBSN with all necessary information to get required data; (2) **User logs in to a WBSN**, in which a user in the role of a RP logs in to a WBSN and accesses to his profile, contacts and resources; and (3) **User accesses to a contact's data**, a user, also in the role of a RP, tries to access the profile and resources of a contact who is registered in a different WBSN.

It should be noticed that analogous to many web applications in which personal data is managed, communications between entities are carried out through a confidential and authenticated channel which also provides data integrity, such as SSL.

7.3.2.1 Initialization

The initialization phase focuses on preparing entities with all required information. It is also divided in three different steps: registration of entities, registration of resources and identity data and specification of main information in WBSNs.

Registration of entities. The registration of entities involves the registration of a Host at an AM and the registration of an IdP at an AM, which can be the same AM or a different one. In particular, these registrations are equivalent to the introduction of a Host to an AM described in UMA [1]. The key point is the establishment of a trust relationship between a Host or an IdP and an AM. This is carried through the participation of a user in the role of an AU. He introduces the Host or the IdP in the chosen AM to make available later validation of tokens.

Notice that an AU can perform this registration more than once. It is possible that, as mentioned in Section 7.2.1, each AU chooses several Hosts, AMs and IdPs.

To conclude, the registration process finishes when the user specifies in his AMs and IdPs the list of trusted IdP_CAs.

Registration of resources and identity data. This phase focuses on registering new resources and the appropriate FOAF file in the selected Host and IdP. Specifically, the registration of resources and identity data is equivalent to the analogous part of UMA [1]. Once again each user takes the role of an AU. The main point is the update of a resource in a Host and the update of the FOAF file in an IdP, together with their later registration in chosen AMs and the establishment of access control policies.

Specification of main information in WBSNs. Once a user enrols for the first time in a WBSN, the specification of the IdP in which his FOAF file is stored and the Host which stores his resources is indispensable. Consequently, the WBSN, once a user is logged or a request from other WBSN is received, is able to access

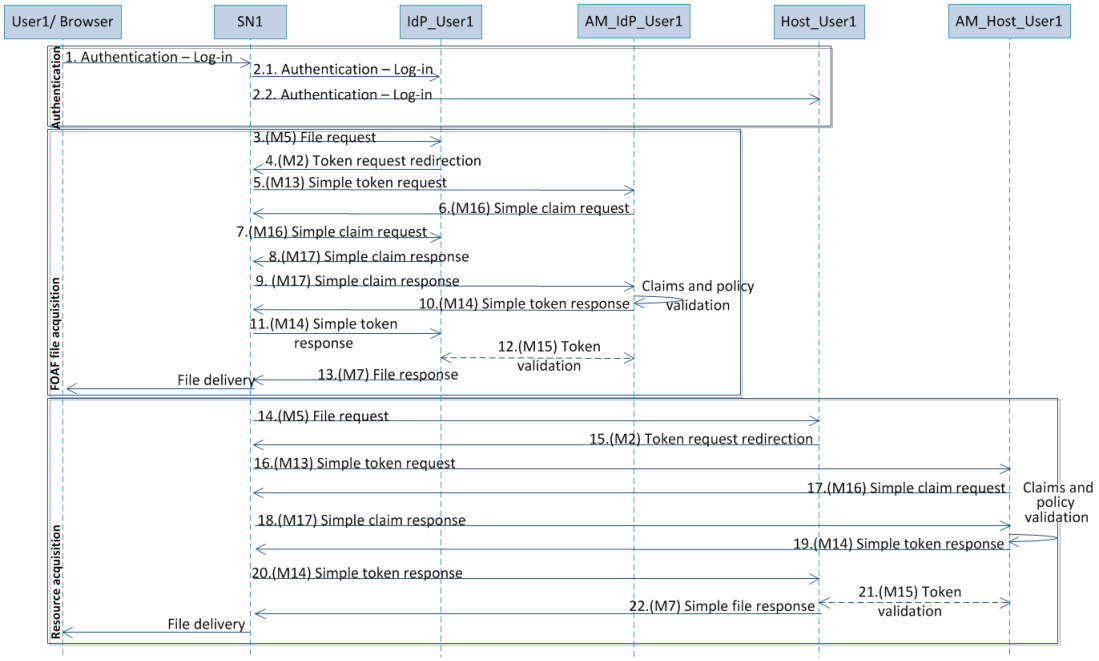


Figure 7.4: User logs in to a WBSN

to the user profile, including personal attributes and contacts, as well as, to user resources.

7.3.2.2 User logs in to a WBSN

Each time a user logins in a WBSN, taking the role of a RP, three processes are carried out, his authentication, the acquisition of his profile and contacts and the acquisition of his resources which remain accessible. In order to acquire these data, the user is authenticated against his IdP and Host by the WBSN and, then, each WBSN, in the role of a requester and on behalf of the user, contacts to user's IdP and Host to get his FOAF file and resources respectively. The step of accessing a protected resource of UMA protocol [1] is executed a couple of times, one to get the FOAF file and another to acquire resources. This process can be performed repetitively and, even more, it can be used to achieve multiples resources in a single execution. It is depicted in Figure 7.4 and each message identifier is pointed out in brackets regarding Table 7.1.

The process requires the acquisition of claims and the necessary mutual authentication between the RP and his Host and IdP (message 1, 2 of Figure 7.4) to later delegate access to WBSNs, being these issues not detailed in UMA. Authentication can be carried out applying multiple mechanisms and protocols. Using symmetric cryptography, some mechanisms in what concerns the Challenge-Response protocol are a feasible choice. By contrast, though increasing complexity, public key cryptography is another alternative, for example using the mutual authentication mechanism proposed by [161]. However, guaranteeing that WBSNs do not impersonate a user requires the performance of the authentication procedure in the logging in and out of the WBSN to inform the user's IdP and Host that he is connected or not. Furthermore, also trying to prevent this issue, all signatures carried out by IdPs and WBSNs, that are mentioned along the paper, include a time stamp. Therefore, users, together with access control policies, specify an accepted time stamp threshold. Also, notice that IdPs and AMs acquire the time from a trusted site such as the NIST Internet Time Service [162].

The login phase starts requesting the user's FOAF file to IdP_User1 (msg. 3). Due to the need of getting an access token, the request is redirected to AM_IdP_User1 which requests claims before providing the token (msg. 4-6). Claims correspond to a proof to identify the RP. In order to acquire them, the WBSN in which the RP delegates requests the accreditation of the RP to IdP_User1 (msg. 7). Then, this IdP provides requested claims, that is, a signed structure composed of a reduced FOAF file with the name and email of the user (msg. 8). Subsequently, when AM_IdP_User1 receives claims, it verifies the signature, making use of the list of IdP_CAs specified by the user, and validates access control policies to later deliver the appropriate token (msg. 9). Finally, the token is presented to IdP_User1 and the requested file is lastly delivered (msg. 10-13). Once claims and tokens are obtained, they are stored in the WBSN for the whole session of the user. Then, if needed, they are delivered without having to be requested again. Nonetheless,

the erasure of tokens and claims when a user logs out is required to prevent user impersonations.

On the other hand, user's resources are requested following the same procedure though without requesting claims but reusing (msg. 14-22). Indeed, claims are stored in WBSNs along each user session to be repetitively used until they expire if they do so.

As a final remark, according to messages' content pointed out in Table 7.1, the verification of signed messages requires to check if signing entities are or not within the established lists of IdP_CAs and WBSN_CAs. In case that signing entities are not in the lists, the protocol is aborted. The dynamic enlargement of lists through the inclusion of new entities is left for future work.

7.3.2.3 User accesses a contact's data

Once a user is within a WBSN in multiple circumstances desires to access data of his contacts. However, if contacts are enrolled in different WBSNs, interactions between these applications are indispensable. First of all, given a user of WBSN1, User1, which wants to access resources of one of his contacts, User2, all WBSNs in which User2 is registered in have to be identified. Indeed, this information is available in the FOAF file of User1, as described in Section 7.2.3. Then, User1 chooses one WBSN, for example WBSN2, and the procedure described in this Section is performed.

When User1 desires to visualize the profile and resources of User2, he access User2 relationship, and if this user also has a relationship with User1, his profile and resources are delivered according to User2's access control policies.

This process is composed of a pair of executions of the step of accessing a protected resource within UMA protocol. One execution is carried out to acquire the reduced FOAF file of User2. As this file is reduced, contact relationships of User2 are not included and attributes are available according to access control

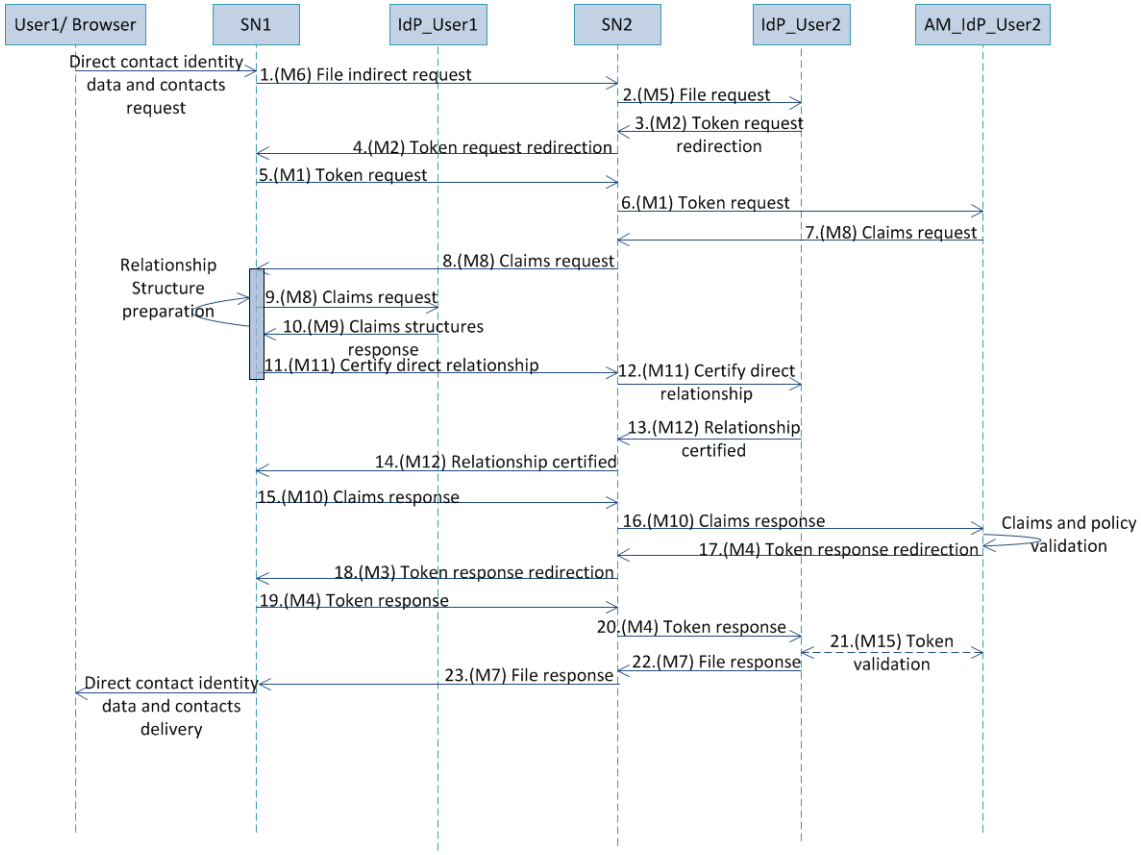


Figure 7.5: User accesses a to contact FOAF file

policies. The second UMA execution corresponds to the acquisition of resources of User2 and it can be performed repetitively.

For the sake of brevity only and due to the analogy between acquiring the profile and resources of User2, which only differs on requesting data to an IdP or to a Host, the following Section presents the acquisition of User2' FOAF file. The process is depicted in Figure 7.5 and details, in brackets, messages identifiers regarding Table 7.1.

FOAF file acquisition The procedure differs from UMA in a couple of points. On the one hand, WBSN1 and WBSN2 play together the role of a Fat Requester, as pointed out in Section 7.2.

On the other hand, claims are clearly detailed. In particular, to obtain the

token that grants access to requested identity data, the AM_IdP_User2 requests claims to User1 that consist of three elements. The first element corresponds to a proof that certifies the existence of a bidirectional relationship between User1 and User2, such that (explained in Section 7.1) User1 is in User2's FOAF file and on the other way round (P1). Therefore, assuming that User2 is already in the FOAF file of User1, the proof refers to a relationship structure regarding the existence of User1 relationship in the FOAF file of User2. The second element corresponds to a proof of possessing some attributes. It is a structure that depends on access control policies, thereby attributes can be requested or not and they can differ from one request to another (P2). The last proof corresponds to the identification of the RP, User1 (P3).

More specifically, User1 can request User2's FOAF file or delegate in WBSN1. Supposing that User1 delegates in WBSN1, this WBSN requests to WBSN2 User2's FOAF file which redirects the request to IdP_User2 (messages 1, 2 of Figure 7.5). Next, this entity also redirects to AM_IdP_User2 to get the access token (msg. 3-6). However, AM_IdP_User2 requests claims before delivering the token (msg. 7, 8). It should be noticed that structures involved in *Claims request* are empty except for requested attributes. In particular, claims corresponds to a signed structure in relation to requested attributes (P1), a signed relationship structure which identifies the relationship between both users (P2) and a signed structure to certify User1's identity (P2). Consequently, WBSN1 acquires claims through IdP_User1 and IdP_User2. First, IdP_User1 delivers all the three structures. Afterwards, the last pair of them are sent to WBSN2 and redirected to IdP_User2 (msg. 9-12) to verify the existence of a relationship between User2 and User1 (User1 in User2's FOAF file). When IdP_User2 successfully performs the appropriate verifications, it signs the received relationship structure, introducing a time stamp, and sends it back to WBSN1 (msg. 13, 14). Next, WBSN1 sends claims to AM_IdP_User2 (msg. 15, 16). Finally, the access token is delivered and presented to IdP_User2

which provides the requested FOAF file (msg. 17-23).

Eventually, there are some points to remark. Firstly, the acquisition of resources and profiles can be performed multiple times and depends on access control policies attached to them. In case many data are joined under the same policy, the token obtained provides access to all of them. On the contrary, if each resource has a different access control policy attached to it, the token acquired would only provide access to a single piece of data. Secondly, as pointed out in Section 7.3.2.2, claims are stored in the WBSN that initially sends the request to, if required, be later delivered without the need of requesting them again. Similarly, if the procedure of achieving data, either resources or identity data, is executed repetitively in a WBSN, tokens are stored in the WBSN that initially sends the request and they may remain valid, thereby reused, if their expiration times are not exceeded. Then, in such cases the procedure is simplified since messages to get tokens are not required.

7.4 Summary of the chapter

This Chapter has presented U+F, a protocol to achieve interoperability and reusability of identity data, resources and access control policies among different WBSNs. It is based on a pair of proposals, mainly, UMA protocol and the FOAF project. Furthermore, access control management is carried out concerning a simplified version of $SoNeUCON_{ABC}$ usage control model.

Chapter 8

Extended UMA+FOAF Social Network Protocol. Including data exposure minimization and indirect relationships management

---

Given the quantity and assorted purposes of WBSNs, users want to interact with people no matter the WBSN in which they are enrolled, thereby attaining interoperability and reusability according to resources, identity data and access control policies.

Moreover, most of WBSN users look for new people to whom establish some kind of relationship, without necessarily being direct contacts. Indeed, indirect relationships are an inherent property of the society. As C. Calhoun noticed [163], society is a question of social integration where the growing relevance of *indirect relationships* is related to modernity. Indirect relationships in WBSNs correspond to the number of jumps that users can perform from one user to others, also called depth [28, 148], and their establishment is essential. Furthermore, to attain fine-grained access control the establishment of indirect relationships is indispensable because a significant number of WBSN features are associated with them, namely,

distance, *multi-paths* and *common-contacts* (recall Section 2.4).

Other desirable feature is the protection of data against unnoticed or non-consented uses. There have been several attempts to conceal data from servers [164, 165] and, regarding recent trends, it is referred as *data exposure minimization* (recall Section 2.4). In the great majority of cases, when registering in a WBSN the acceptance of the established privacy policy is mandatory. Multiple WBSN privacy policies specify the management and use of all uploaded data. An extremely related example is the new Google's privacy policy in which the use of all users' data to improve experience in Google applications is detailed [166].

U+F achieved interoperability and reusability regarding direct relationships between WBSNs combining the application of UMA + FOAF (Chapter 7). In order to address the remaining couple of features, particularly, data exposure minimization and indirect relationships management across different WBSNs, this Chapter presents the Extended UMA + FOAF Social Network protocol (eU+F). It combines, under the bases of $SoNeUCON_{ABC}$, the application of UMA and FOAF, together with cryptographic techniques.

This Chapter is structured in the following Sections. At a starting point, an overview of the system is presented in Section 8.1. In Section 8.2 the system model is described, including the system architecture, the requirements, and the trust and adversarial models. Next, Section 8.3 describes the protocol. In Section 8.4 developed cryptographic schemes to deal with data exposure minimization are detailed. Finally, changes to improve eU+F access control management regarding a full consideration of $SoNeUCON_{ABC}$ are discussed in Section 8.5.

8.1 System overview

U+F achieves interoperability and reusability among different WBSNs (Chapter 7). However, a pair of demanding necessities are out of the scope of U+F. First, as in current WBSNs, indirect relationships have to be managed. Second, data is out

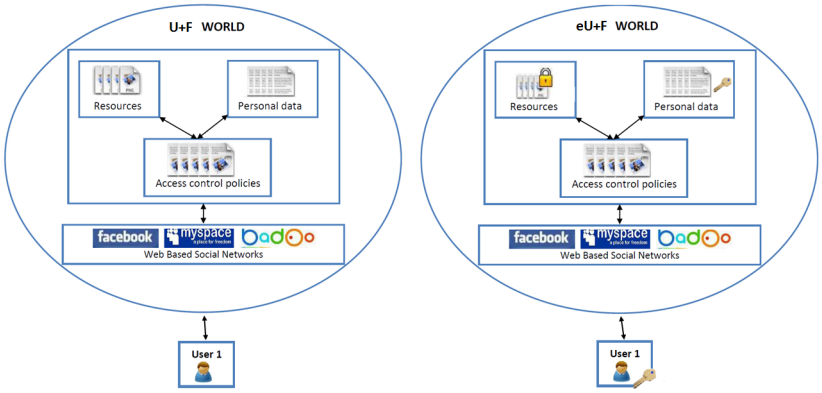


Figure 8.1: U+F vs eU+F

of users full control as WBSNs can use them for their own purposes without users consent. As a result, in order to face up these new challenges, in this Chapter a more powerful and secure protocol is proposed, the Extended UMA+FOAF Social Network Protocol (eU+F) (see Figure 8.1).

From a more specific point of view, like in U+F, identity data corresponds to the profile and contacts of each user, and they are stored in the form of FOAF files within IdPs; resources are stored encrypted in chosen Hosts; and access control policies are located in AMs, which perform access control on behalf of the users. However, the eU+F differs from U+F in that that resources and identity data are delivered encrypted and they have to be decrypted in the users browsers. Moreover, apart from the phases applied in U+F (**Initialization**, **User logins in a WBSN** and **User accesses data of a direct contact**), eU+F involves a new phase, **User accesses data of an indirect contact**. It consists of accessing to an indirect contact enrolled in another WBSN (different from any other). It is divided in the acquisition of identity data and resources and it works similarly to the access to a direct contact data, but requiring a proof to verify the existence of the appropriate indirect relationship.

This extended version, like U+F, manages access control on a simplified version of $SoNeUCON_{ABC}$ where the rt structure is not constructed in order to verify access control policies. Instead, the alternative approach mentioned in Section 4.4

of this thesis is adopted: "The first one consists of searching for enriched paths between the administrator a of the requested object o and the requester s throughout the whole WBSN graph (G), and verifying the policies during the process". Furthermore, in eU+F, it is assumed that users access data of indirect contacts that can be reached from their direct contacts, that is, this is how (indirect) contacts are discovered. Moreover, recalling features managed within $SoNeUCON_{ABC}$ usage control model, namely, *direction*, *distance*, *common-contacts*, *multi-path*, *clique*, *flexible attributes* and according to the proposed policy language (Section 4.3), eU+F focuses only on indirect relationships management and *cliques*, *common-contacts* and *multi-paths* are not considered. Nonetheless, eU+F can be extended to deal with all of these features, as well all as to manage usage control, being both issues discussed in Section 8.5.

In the following, the way that eU+F proceeds is illustrated with an example (see Figure 8.2). Analogously to U+F, assuming a direct relationship between U_2 and U_1, and the fact that U_1 wants to access U_2's data, the access is granted if the relationship is bidirectional and a proof of the existence of the relationship U_2-U_1 is obtained from IdP_U_2, that is, U_1 is within U_2's contacts (solid arrow). On the other hand, given the management of indirect relationships proposed in eU+F, supposing that U_1 has already accessed to U_2's profile (including his direct contacts) and U_1 wants to access U_4's data, the access is granted if there exist bidirectional relationships between all involved users in the path and it is obtained from IdP_U_4 a proof of the existence of a relationship between U_4-U_1. This proof is constructed step by step. First, IdP_U_3 certifies the relationship U_3-U_1 (solid arrow) and then, after presenting this proof to IdP_U_4, this IdP certifies the relationship U_4-U_3 (solid arrow). Finally, the proof U_4-U_1 is constructed. Therefore, it is clearly noticed that access control focuses on the existence of relationships in the opposite direction to the discovery of contacts. However, it is remarkable that getting the proof is not enough to get access because it depends on access control policies and thus, not only

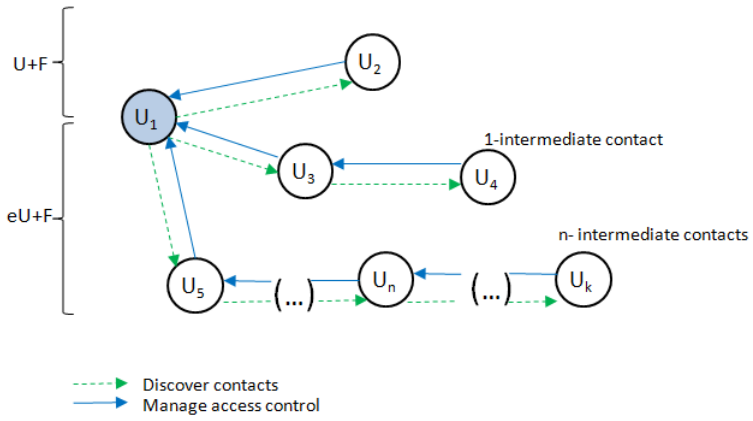


Figure 8.2: Managed relationships

the proof has to be obtained but also policies have to be satisfied. Finally, note that the set of relationships that are managed in eU+F are the most practical because, for efficiency reasons, instead of constructing the rt structure, the enforcement process is performed constructing a path, jumping from a direct contact to another, between the administrator and the requester and verifying policies accordingly.

8.2 System model

The model involves the specification of the architecture (Section 8.2.1), the system requirements (Section 8.2.2) and the trust and adversarial model (Sections 8.2.3 and 8.2.4 respectively).

8.2.1 Architecture

The architecture of eU+F, depicted in Figure 8.3, is analogous to that of U+F (Section 7.2.1) except for the inclusion of a new group of Certification Authorities (ACs) and the addition of new tasks. Differences between both protocols in respect to involved entities are the following:

1. *User*: each user is in charge of creating, at least, a symmetric key used in the encryption and decryption of resources and an asymmetric key pair (which

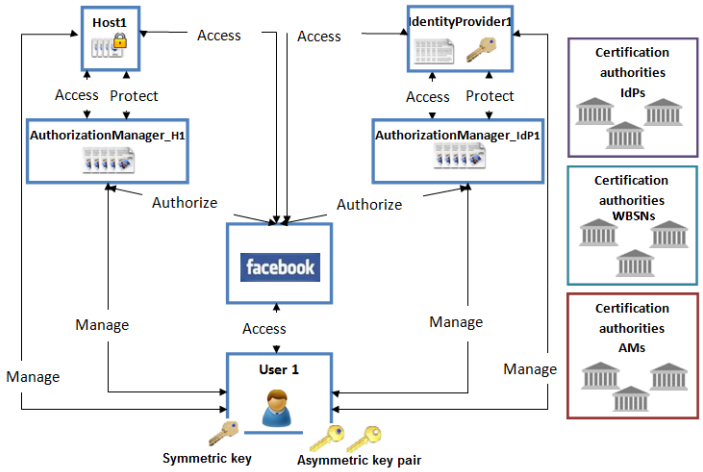


Figure 8.3: eU+F architecture

can correspond to a private key and the associated public key certificate or to a created key pair) to manage decryptions and interchanges of encrypted data. These keys are used in the schemes described in Section 8.4 to attain data exposure minimization.

2. *Identity provider (IdP)*: apart from being repository of FOAF files, in eU+F, decryption keys are stored.

3. *Host*: repository of encrypted resources which have to be periodically re-encrypted, either by the host under the users' supervision or directly by users who update the data re-encrypting it with a new key. Note that re-encryptions require the update of the used key in the appropriate IdP.

4. *Authorization Manager (AM)*: this entity owns a certificate and the associated private key to sign claims. Then, trustworthiness of requested claims is guaranteed.

5. *Social Network*: assuming the need of decrypting identity data and resources, WBSNs must provide the appropriate procedures to allow decryptions at the client-side, that is, at users' browsers.

6. *Certification authorities* (CA): a new group of certification authorities (CA), for AMs, is added (AM_CA).

8.2.2 Requirements

Regarding eU+F features, apart from requirements pointed out in U+F (Section 7.2.2), that refer to interoperability and reusability regarding direct relationships, resources and identity data confidentiality and access control, resources and identity data integrity, chain of trust and access minimum identity data, the following requirements are challenges to attain:

1. **Interoperability and reusability regarding indirect relationships.** The communication and interchange of data between multiple users enrolled in different WBSNs has to be attained. Moreover, both features have to be provided in regard to indirect relationships.

2. **Resources and identity data protection against WBSNs.** Resources and identity data have to be adequately protected from WBSNs. It is subdivided in the following requirements:

   (a) **Resources and identity data exposure minimization.** Resources and identity data have to remain inaccessible to WBSNs, being protected against inappropriate managements. Furthermore, it is desirable that Hosts do not get access to stored resources.

   (b) **Access minimum identity data**. As in U+F, accessible identity data among WBSNs has to be minimized. In particular, indirect relationships have to remain, as much as possible, unknown for WBSNs.

3. **Simple key management**. Keys have to be easily managed, which means that decryption keys are not distributed out of band such as it is done in [70] or in [62] because, due to the large amount of users, the distribution can become unmanageable.

8.2.3 Trust model

The trust model is based on the following assumptions:

- IdPs and AMs are trusted entities. Thus, they do not maliciously manipulate data and they guarantee no protocol deviation.
- Hosts and WBSNs are untrusted entities. They both, can use resources and identity data for chosen purposes and produce protocol deviations.
- CAs are in charge of managing trust, that is, they issue certificates to trusted IdPs, WBSNs and AMs.

8.2.4 Adversarial model

Additionally to U+F whose adversarial model consists of adversaries that can inject, alter and get data interchanged along the protocol, as well as adversaries that can impersonate IdPs and WBSNs (Section 7.2.5), in eU+F a harder adversary is assumed. In particular, in this extended protocol it is considered that adversaries may try impersonating AMs and can get access to tokens immediately after being delivered by AMs.

8.3 eU+F protocol description

eU+F is described in terms of messages content (Section 8.3.1) and the execution process (Section 8.3.2).

8.3.1 Messages content

Messages content is equivalent to U+F but adding cryptographic *operations* and modifying the *RelationshipA-B structure*. Besides, some considerations in regard to tokens *elements* should be added.

8.3.1.1 Operations

Due to the use of cryptography to deal with data exposure minimization, symmetric and asymmetric algorithms are applied according to the schemes proposed in Section 8.4.

8.3.1.2 Elements

In what concerns tokens, their use and application remains being analogous to U+F (recall Section 8.3.1.2). Nonetheless, recalling the problem of misusing tokens to impersonate users, this issue is avoided as data is delivered encrypted. Then, even if WBSNs get unauthorized access to data, they could not decrypt them and impersonations would not affect users privacy.

8.3.1.3 Structures

In this extended protocol, the structure *RelationshipA-B* used in U+F has to be modified. In particular, *RelationshipA-*B_i is applied instead and it refers to the identifiers of the users involved in the relationship, where i refers to the number of jumps that separate both users. Besides, this structure is defined as *first: hash_email_Requester*, *end: hash_email_Administrator* and *depth: value*. Notice that depth is used to considering the depth of a relationship in case of indirect relationships and due to theoretical studies [153] that explain that any two people on this planet can be connected via an average number of six steps, the depth of indirect relationships is bounded to seven users (six the maximum depth).

Messages interchanged in the eU+F protocol are depicted in Table 8.1 where symbol $\|$ implies concatenation, S means signature and E means encryption. This Table presents each message content in regard to operations, elements and structures previously described in Section 8.3.1.

Table 8.1: Interchanged messages in eU+F

Id	Name	Content
M1	Token request	Ticket $\|WBSN_R\_Cert\_Serial\_Number\|$ Date_time$_{WBSN_R signature}$|S$_{k_{WBSN_R\text{-}Cert}}$(Complete Message)
M2	Token request redirection	Ticket $\|AM\_location$
M3	Token response redirection	Ticket $\|Tokenvalue\|Expired-in\|$ AM$_A_Cert_Serial_Number\|Date_time_{AM_A signature}\|$ $S_{k_{AM_A\text{-}Cert}}$(Complete Message)
M4	Token response	Token response redirection $\|WBSN_R\_Cert\_Serial\_Number\|$ Date_time$_{WBSN_R signature}$|S$_{k_{WBSN_R\text{-}Cert}}$(Complete Message)
M5	File request	R_Id $\|A\_Id\|File\_Id\|WBSN_R\_Cert\_Serial\_Number\|$ Date_time$_{WBSN_R signature}$|S$_{k_{WBSN_R\text{-}Cert}}$(Complete Message)
M6	File indirect request	R_Id $\|A\_Id\|File\_Id\|WBSN_R\_Cert\_Serial\_Number\|$ Date_time$_{WBSN_R signature}$|S$_{k_{WBSN_R\text{-}Cert}}$(Complete Message)
M7	File response	R_Id $\|A_Id\|$E$_{k_R}$(file)
M8	Claims request	R_Id $\|A_Id\|$E$_{k_{CertIdP_R}}$(Data R request) $\|$ AM$_A_Cert_Serial_Number\|Date_time_{AM_A signature}\|$ $S_{k_{CertAM_A}}$(Complete Message)
M9	Claims structures response	R_Id $\|A\_Id\|AccreditationR\|$ $E_{k_{CertAM_A}}$(Data R response)$\|IdP_R\_Cert\_Serial\_Number\|$ Date_time$_{IdP_R signature}$|S$_{k_{IdP_R}}$(Complete Message)
M10	Claims response	Claims structures response $\|RelationshipR\_A_1\ \|$ IdP$_A_Cert_Serial_Number\|Date_time_{IdP_A signature}\|$ $S_{k_{IdP_A}}$(Relationship R_A$_1$) $\|WBSN_R\_Cert\_Serial\_Number\|$ Date_time$_{WBSN_R signature}$|S$_{k_{WBSN_R\text{-}Cert}}$(Complete Message)
M11	Certify direct relationship	R_Id $\|A\_Id\|AccreditationR\|$ IdP$_R_Cert_Serial_Number\|Date_time_{IdP_R signature}\|$ $S_{k_{IdP_R}}$(Accreditation R)$\|RelationshipR\_A_1\ \|$ WBSN$_R_Cert_Serial_Number\|Date_time_{WBSN_R signature}\|$ $S_{k_{WBSN_R\text{-}Cert}}$(Complete message)
M12	Certify indirect relationship	R_Id $\|A\_Id\|AccreditationR\|$ IdP$_R_Cert_Serial_Number\|Date_time_{IdP_R signature}\|$ $S_{k_{IdP_R}}$(Accreditation R) $\|RelationshipR\_A_i\ \|$ IdP$_{A_i}_Cert_Serial_Number\|Date_time_{IdP_{A_i} signature}\|$ $S_{k_{IdP_{A_i}}}$(Relationship R_A$_i$) $\|RelationshipR\_A_1\ \|$ WBSN$_A_Cert_Serial_Number\|Date_time_{WBSN_A signature}\|$ $S_{k_{WBSN_A\text{-}Cert}}$(Complete message)
M13	Relationship certified	R_Id $\|A\_Id\|RelationshipR\_A_1\ \|$ IdP$_A_Cert_Serial_Number\|Date_time_{IdP_A signature}\|$ $S_{k_{IdP_A}}$(Relationship R_A$_1$)
M14	Simple token request	Ticket
M15	Simple token response	Ticket $\|Tokenvalue\|Expired-in\|$ AM$_A_Cert_Serial_Number\|Date_time_{AM_A signature}\|$ $S_{k_{AM_A\text{-}Cert}}$(Complete Message)
M16	Simple file request	R_Id $\|File\_Id$
M17	Simple file response	R_Id $\|$E$_{k_R}$(file)
M18	Simple claim request	R_Id $\|A\_Id$
M19	Simple claim response	R_Id $\|IdP_R\_Cert\_Serial\_Number\|AccreditationR\|$ Date_time$_{IdP_R signature}$|S$_{k_{IdP_R}}$(Accreditation R)
M20	Token validation	Ticket $\|Tokenvalue$

8.3.2 Execution procedure

eU+F is divided in four phases: (1) the **initialization** phase, in which the initialization of entities is performed; (2) **User logs in to a WBSN**, in which a user, in the role of a RP, logins in a WBSN and accesses to his encrypted identity data and resources, being data locally decrypted; (3) **User accesses to data of a direct contact** where a user, also in the role of a RP, tries to access the profile and resources of a direct contact who is registered in a different WBSN, being data locally decrypted; and (4) **User access data of an indirect contact** in which a user, again in the role of a RP, accesses to data of an indirect user who is registered in a different WBSN (data is also locally decrypted). It is remarkable that accessing a direct or an indirect contact data within the same WBSN follows the same procedure as accessing data of a user enrolled in a different one.

It should be noticed that phases (2) **User logs in to a WBSN** and (3) **User accesses to data of a direct contact** only differs from U+F in the following pair of issues and thus, they are not described.

- Identity data and resources, once obtained, have to be decrypted at users' browsers following one of the schemes described in Section 8.4

- Requested claims and delivered tokens are signed by appropriate AMs and signatures verify accordingly.

- A new structure is applied in claims management, *RelationshipA-*B_i, that involves the element *depth* to deal with indirect relationships.

By contrast, some tasks are added in the **initialization** phase (Section 8.3.2.1) and the **User access data of an indirect contact** phase is described from scratch (Section 7.3.2.1).

8.3.2.1 Initialization

As an additional task, users have to create a set of keys. Moreover, the specification of lists of trusted WBSN_CAs in AMs, trusted AM_CAs and WBSN_CAs in IdPs and trusted WBSN_CAs in Hosts is required. Furthermore, users have to store in chosen Hosts his resources, encrypted, and the symmetric keys applied in the resources encryption in their IdPs.

8.3.2.2 User accesses data of an indirect contact

Considering the existence of indirect relationships, the procedure is rather similar to access data of a direct contact except for requiring interactions between all WBSNs involved in the relationship. In particular, WBSN interactions are indispensable to acquire claims that prove the existence of an indirect relationship between a pair of users. For instance, given three users such that User1 is directly connected to User2 and User2 to User3, to verify the indirect relationship between User3 and User1 a proof of the existence of such relationship is requested to IdP_User3. Then, the request sent to IdP_User3 attaches a proof of the relationship between User2 and User1 and IdP_User3 verifies if User3 has a relationship with User2 to finally certify the indirect relationship between User3 and User1. Nonetheless, it is noteworthy that apart from getting the proof, User3's access control policies have to be satisfied to get the requested access.

As in *User accesses to a contact's data* (Section 7.3.2.3), the procedures of acquiring identity data and resources are quite analogous and the main difference is that IdPs provide identity data and Hosts provide resources. Thus, given the previous example, the following Section describes the acquisition of User3's FOAF file. It is depicted in Figure 8.4 and details, in brackets, messages identifiers regarding Table 7.1.

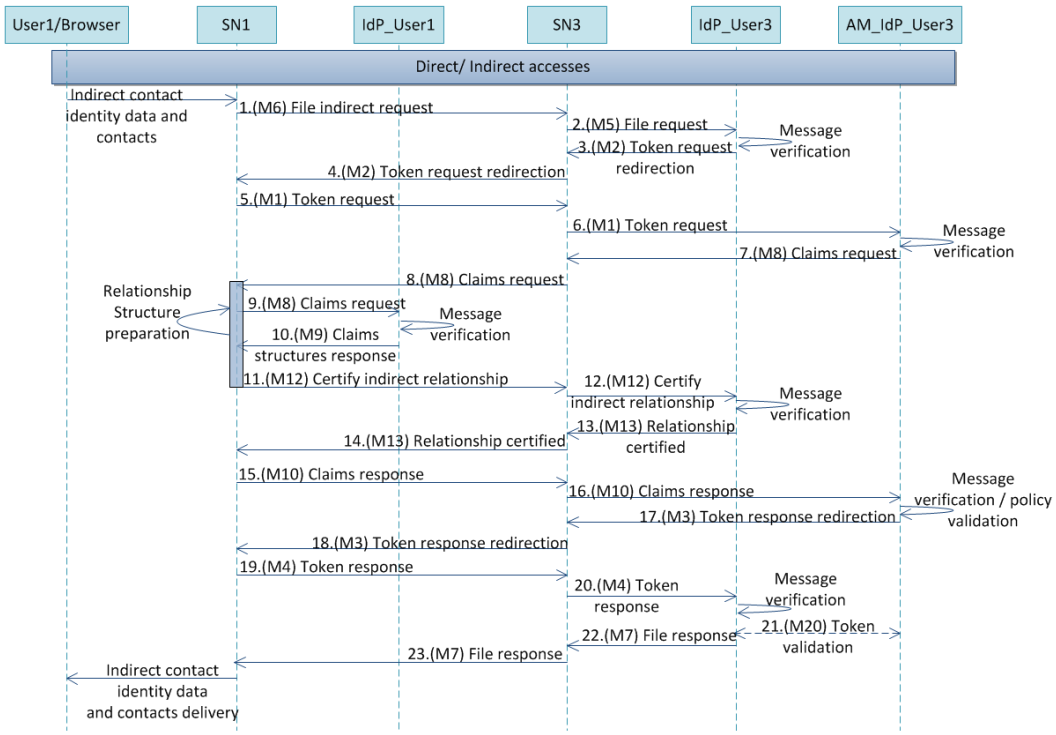


Figure 8.4: User accesses to the FOAF file of an indirect contact

FOAF file acquisition The acquisition of a FOAF file requires as many UMA executions as WBSNs are involved in the relationship minus one. In general, this procedure focuses on recursively repeating the acquisition of FOAF files from each of the WBSNs involved the indirect relationship. Indeed, this feature is what points out that an indirect relationship can be defined as multiple direct ones. According to the proposed example two execution of UMA are performed, the first one to get the FOAF file of User2 and the second one to get the FOAF file of User3. The procedure is depicted in Figure 8.4 and, in brackets, message identifiers in regard to Table 8.1 are noticed.

More specifically, after having acquired identity data of User2, applying the procedure described in Section 7.3.2.3, User1 clicks on User2 relationship with User3. Afterwards, User1 chooses to access User3's identity data and an the protocol described next is carried out. WBSN1 interacts with WBSN3 and it requests to IdP_User3 the User3's FOAF file (msg. 1, 2 of Figure 8.4). Subsequently, IdP_User3

requests an access token and redirects WBSN1 to AM_IdP_User3 (msg. 3-6). Then, AM_IdP_User3 requests claims (msg. 7, 8) that are analogous to the ones requested when accessing a direct contact except for P2 which has to proof the existence of the indirect relationship between User3 and User1. Therefore, P1 is reused and P3 is reused or requested depending on requested claims (msg. 9, 10). By contrast, obtaining P2 requires the interaction with WBSN3. Indeed, WBSN1 creates P2* that corresponds to a non-certified proof of the relationship between User3 and User2 and sends it together with the P2 previously obtained (while accessing to User2's data) that certifies the relationship between User2 and User1 to IdP_User3 (msg. 11,12). The IdP_User3 verifies the existence of the relationship, creates the new P2 and sends it back (msg. 13,14). When WBSN1 gets claims (composed of P1, P2 and P3), sends them to AM_IdP_User3 and if their verification is successful the access token is delivered (msg. 15-18). Lastly, the token is sent to IdP_User3 and the requested file is provided (msg. 19-23). Once again, the IdP delivers an encrypted reduced FOAF file that has to be decrypted in the user's browser applying one of the schemes proposed in Section 8.4.

Finally, as in other phases, each signed message has to be verified. Thus, if signing entities are within the established lists to abort the protocol otherwise has to be identified.

8.4 Data exposure minimization management

There are multiple possibilities, making use of cryptography, to prevent WBSNs from visualizing resources and identity data presented in them. However, regarding one of the security requirements, decryption keys cannot be distributed off-line because, as WBSNs are used by a huge quantity of users and lots of them are not directly known, the procedure would be impractical. Therefore, an hybrid encryption approach, similar to [167], is applied to resources management and an asymmetric one to identity data management. In particular, a pair of alternatives

to manage and distribute keys are described in the following Sections, one of them focuses on traditional Public Key Cryptography (PKC) and the other one focuses on PKC based on Identity Based Encryption (IBE). However, the election of a particular algorithm is an open issue. Moreover, as users' emails are considered subjects attributes, Attribute Based Encryption (ABE) could have been applied instead of IBE.

Furthermore, the analysis of advantages and disadvantages of achieving data exposure minimization is essential. The main advantage is to prevent WBSNs from using personal data for their own purposes such as sending spam or building profiles of users likes and dislikes. Nonetheless, there are some drawbacks to highlight. Firstly, the time required to perform the protocol increases due to the cryptographic operations applied. Second, users are in charge of encrypting their resources and uploading them and the applied key. Third, a particular amount of extra storage is required to store keys. Finally, several messages are added to the protocol, such as those for providing the decryption keys. Next, Sections 8.4.1 and 8.4.2 describe the application of PKC and IBE and Section 8.4.3 presents a comparison of the application of both schemes in eU+F.

8.4.1 Traditional PKC

This technique is based on the well-known concept of PKC [168]. Each user owns a key pair (K_{pub} and K_{pv}), or multiple ones.

In the *Initialization* phase each user delivers his K_{pub} with his FOAF file and his resources decryption key, DK, to the preferred IdP. Then, acquisition of identity data focuses on requesting the appropriate K_{pub} and use it to encrypt and retrieve the requested FOAF file. On the other hand, resources, encrypted with DK, are retrieved and decrypted using K_{pub} to reach DK. The use of this mechanisms involves introducing some new messages apart from those already present in Section 8.3.1. To get a better picture of interchanged messages, Figure 8.5 depicts the ac-

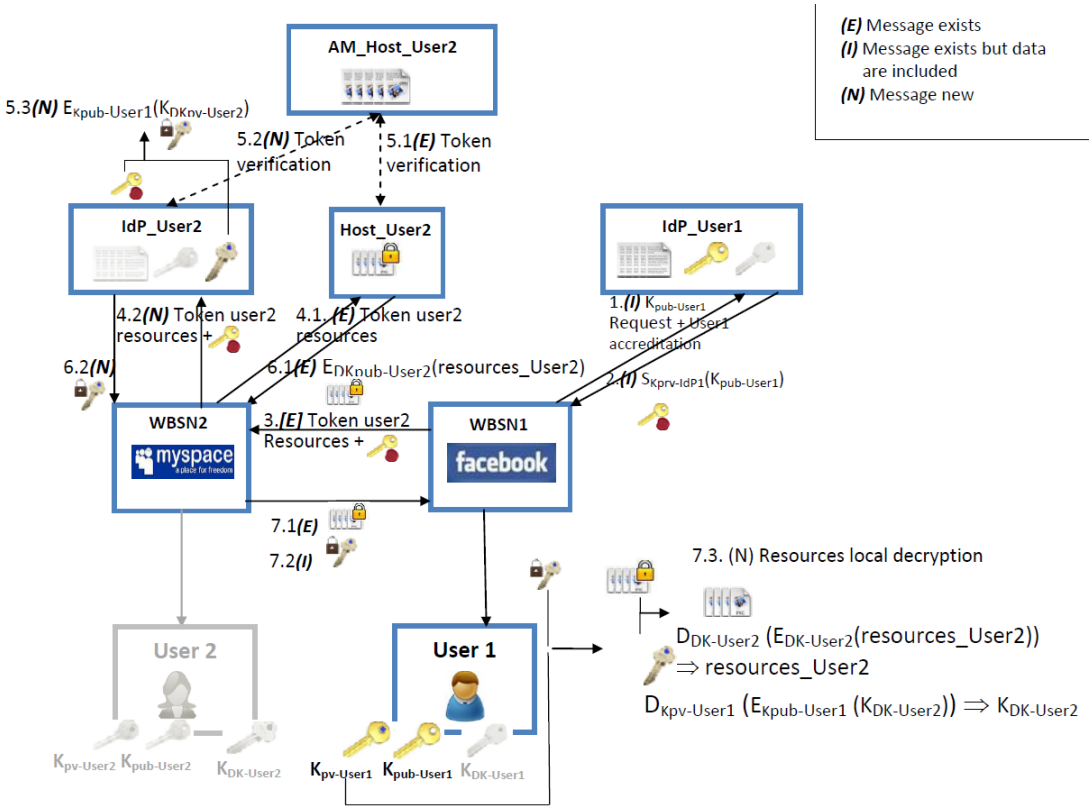


Figure 8.5: Traditional PKC - Acquiring User2's resources

quisition of resources, where (E) points out messages that already exist, (I) refers to messages that are new but can be included within existing ones and (N) points out new messages that have been created from scratch.

8.4.2 IBE-based PKC

To reduce the burden of key management, recalling Section 2.3, Identity Based Encryption (IBE) cryptography is applied [74]. The general approach requires trusted third parties, called herein IBE authorities, to create keys based on some agreed variables. However, the use of an algorithm like [169] is recommendable. It focuses on exclusively creating the public key through public parameters without depending on an additional number chosen by a user or by an authority.

Assuming that eU+F uses the users' email as an identity attribute, once the

Table 8.2: PKC vs IBE-based PKC

	Traditional PKC	IBE-based PKC
Simple key management	√	√
Extra entities		√
Extra storage	√	
Impact performance		√
Key escrow problem		√

attached K_{pv} is provided by an IBE authority, the acquisition of identity data and resources is analogous to the PKC technique except for not requiring the delivery of K_{pub} in IdPs.

8.4.3 Comparison: traditional PKC vs IBE-based PKC

This Section analyses advantages and disadvantages of traditional PKC and IBE-based PKC schemes. A summary of the analysis is presented in Table 8.2.

Both schemes present a pair of significant common advantages in regard to key management. Firstly, on-line key interchanges are not required. This is essential in applications like WBSNs because users share data among multiple contacts and key distribution may be a burden. Second, keys can be periodically updated preventing attacks, in the traditional PKC scheme, against the applied encryption algorithm and in the IBE-based PKC scheme, against the applied IBE algorithm. Indeed, in this last scheme, the update of keys may involve the change of parameters in the used IBE key creation algorithm.

Concerning the traditional PKC scheme, it has the advantage of not involving extra entities in the protocol. Besides, this scheme presents the huge benefit of not affecting the protocols performance to a great extent, that is, resources are symmetrically encrypted and just decryption keys management uses asymmetric cryptography. Moreover, decryption keys are exclusively managed by their creators and the key escrow problem is avoided. By contrast, the IBE-based PKC scheme suffers from the key escrow problem. Besides, apart from involving operations with high computational cost, IBE authorities take part in the protocol. Nonetheless, these new entities release the necessity of extra storage space for public keys, as

well as the necessity of being IdPs in charge of their delivery.

8.5 Modifying eU+F to fully support $SoNeUCON_{ABC}$: a powerful approach

eU+F is based on $SoNeUCON_{ABC}$ and access control has to be performed accordingly. In this regard, although the current development of eU+F only considers indirect relationship, thus managing feature *distance*, it can be extended to manage the remaining set of WBSN features, namely, *direction*, *flexible elements in access control policies*, *cliques*, *distance*, *common-contacts* and *multi-paths* and thus, supporting the definition of any kind of policy applying the $SoNeUCON_{ABC}$ policy language (Section 4.3). Furthermore, eU+F can also be extended to support usage control.

In the following Sections the extension of eU+F to manage all WBSN features and usage control is discussed.

8.5.1 SoNeUCON$_{ABC}$ features management

According to $SoNeUCON_{ABC}$, the management of all WBSN features involves the definition and the evaluation of access control policies defined by the proposed access control policy language (Section 4.3). Access control enforcement can be generally divided into the construction of rt and the later verification of access control policy elements on rt (recall Section 4.4). In this extension of eU+F, this process is carried out by AMs and it is related to claims management.

rt is recursively constructed through the identification of all enriched paths between the requester and the administrator of the requested object (Section 4.4). Assuming that a User1 wants to access to a resource of a User20, the construction of rt should involve all enriched paths between User1 (the requester) and User20 (the administrator). This process is depicted in Figure 8.6, though intermediate nodes have been omitted for brevity. Once the token request message is received

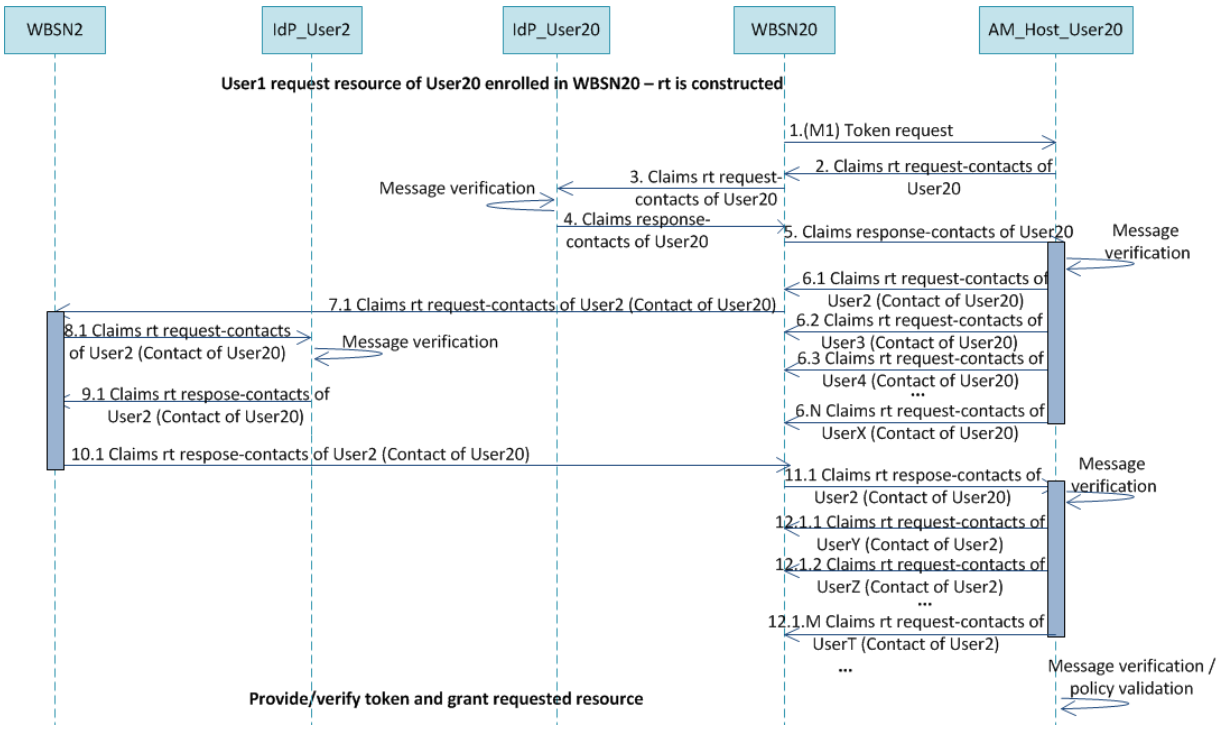


Figure 8.6: eU+F *rt* construction (intermediate nodes omitted for brevity).

by AM_Host_User20, firstly, this AM requests the IdP_User20 all the relationships (E) and the relationships attributes ($ATT(E)$) that User20 owns in his FOAF file (msg. 2-5 Figure 8.6). Subsequently, AM_Host_User20 requests to each IdP of each User20's contact, the relationships and relationship attributes that each of them owns in his FOAF file (msg. 6). For example, AM_Host_User20 starts requesting these data to IdP_User2, being User2 the first contact who User20 owns in his FOAF file (msg. 7-11). Then, this process is recursively performed (msg. 12) until the relationship between User20 and each of the users whose relationship data is requested, has a maximum length of 6 (recall theoretical studies [153]).

After *rt* is constructed, elements involved in access control policies have to be evaluated. In this regard, user, object and relationship attributes ($ATT(S)$, $ATT(O)$ and $ATT(E)$ respectively) are the elements at stake. It should be considered that $ATT(S)$ are included in personal FOAF files and thus, are stored in IdPs. Similarly, some $ATT(E)$ such as $ROLE$ or $TRUST$ are also within FOAF files. By

contrast, other $ATT(E)$ like $DURATION$ may require accessing a Trusted Third Party (TTP), e.g. the NIST Time Service, to properly verify a particular element. On the other hand, $ATT(O)$ are attached to objects stored in chosen Hosts and it is possible that AMs store $ATT(O)$ of all objects to simplify the evaluation of $ATT(O)$. Therefore, the management of all of these attributes, together with the fact that the policy language proposed in $SoNeUCON_{ABC}$ is the one applied herein (recall Section 4.3.3), is directly related to the management of all features and *flexible elements in access control policies* in particular. Applying the proposed policy language all features can be specified in access control policies, being evaluated in rt.

Thus, the verification of policies elements requires some changes in claims management, not only accessing to IdPs but also to Hosts and TTPs. Just the following pair of issues have to noticed:

- Message "*Claims request*" (message 8 Table 8.1) will be larger in size depending on the number of $ATT(S)$ or $ATT(E)$ considered in defined access control policies, that is, ρ_s and ρ_{rt} (recall Chapter 4).

- If a policy involves a ρ_{rt} that considers time values, a TTP should be contacted, so the appropriate AM must make a request to it (Figure 8.7). In a similar way, in the case that access control policies contain ρ_o, the AM should make a request to the adequate Host (Figure 8.7 -A) or, if the AM owns a DB that stores the required object attributes ($ATT(O)$), it may verify ρ_o against it (Figure 8.7 -B).

8.5.2 Attributes and policies management

Particularly, usage control, that refers to permanent management of access control during usage processes, is associated with attributes and access control policies updates, additions and deletions. Once an attribute update, addition or deletion is detected, it is notified to the right AM (the one in charge of managing the

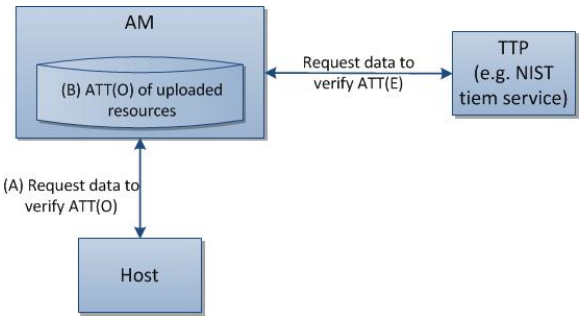


Figure 8.7: eU+F extension to evaluation all kind of access control policies.

updated attribute) to enforce the re-evaluation of access control policies. Likewise, the update, the addition or the deletion of a policy requires the re-evaluation of policies.

Regarding $SoNeUCON_{ABC}$, UDFs are in charge of detecting changes. In relation to attributes management, UDFs are compared with IdPs, Hosts and TTPs. Supposing that User1 is using a requested resource and identity data of User2, the following changes may happen (Figure 8.8): 1) IdP_User2 may change a subject or a relationship attribute, e.g. the user's age; 2) an attribute of the requested resource stored in Host_User2 may change, e.g. the resource's title; 3) in case claims acquisition requires a TTP, it may inform about a change in an attribute, e.g. the timetable; and 4) the intermediate IdPs, that is, the IdP of the requester, may identify a change in an attribute, e.g. a relationship role, and it is informed too. Afterwards, when notifications reach the appropriate AM (AM_User2 in the example), the process of policy evaluation, including rt construction, is repeated.

On the other hand, in case of policy updates, inclusions or deletions, AMs are the entities at stake. These entities act as UDFs, either evaluating the updated or the added policy, or evaluating all access control policies if one of them has been deleted.

It is noteworthy that the repetition of the access control enforcement process is really tedious and striking a balance between privacy and usability is indispensable. For this purpose it could be reasonable to establish a threshold to bound the number

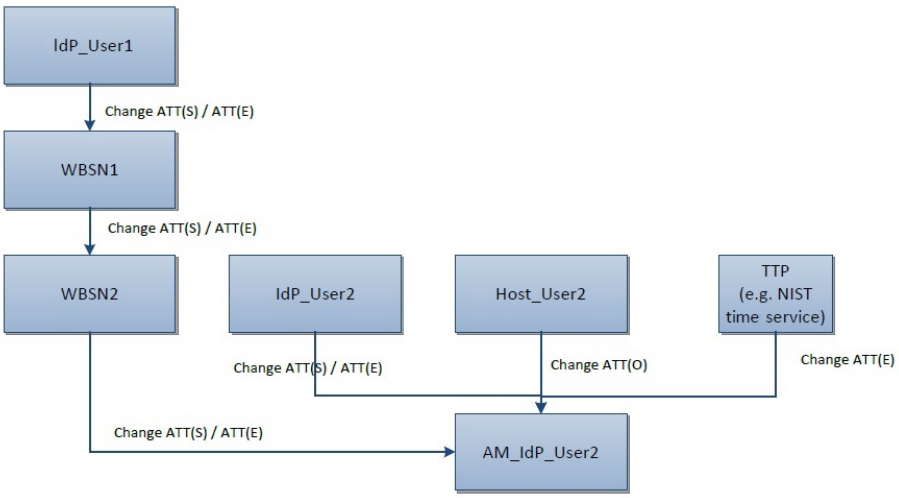


Figure 8.8: eU+F adjustments to manage *changes in attributes*

of times, per resource or identity data request, that access control policies can be evaluated and rt can be constructed.

8.6 Summary of the chapter

This Chapter has presented eU+F Social Network protocol. It is an extension of U+F to, apart from achieving interoperability and reusability, manage indirect relationships and protect data against unnoticed deliveries applying a hybrid cryptographic approach. Furthermore, to fully supports $SoNeUCON_{ABC}$, changes to deal with all WBSN features and usage control are finally discussed.

Part IV

Evaluation and Conclusions

Chapter 9

Evaluation

This Section describes the evaluation of the contributions proposed in this thesis. A graphical summary is depicted in Figure 9.1.

First of all, the evaluation of $SoNeUCON_{ABC}$ shows, theoretically, the significant expressive power of this model by analysing the fulfilment of the set of WBSN features (*distance*, *common-contacts*, *clique*, *multi-path*, *direction* and *flexible elements in access control policies*) and, empirically, the feasibility of its implementation by the development of a proof of concept system which allows the analysis of the access control enforcement process. In addition to $SoNeUCON_{ABC}$, its administrative model $SoNeUCON_{ADM}$ is evaluated by theoretically studying the satisfaction of identified administrative tasks.

The second contribution has been evaluated in three ways. First, a feasibility analysis has been performed over $SoNeUCON_{ABC}$. Second, a prototype has been built to prove the feasibility of implementing CooPeD. Third, a survey has been conducted to assess the usefulness of the proposal.

Finally, the evaluation of U+F and eU+F protocols is presented. Both protocols are theoretically and empirically evaluated. The satisfaction of established requirements and the workload of each protocol's phase is theoretically studied. Moreover, a pair of prototypes prove the viability of implementing U+F and eU+F in a simulated environment. In their regard, the protocol temporal workload has been measured and compared with a pair of popular WBSNs, Facebook and MySpace. Additionally, both protocols have been jointly compared, theoretically and empirically.

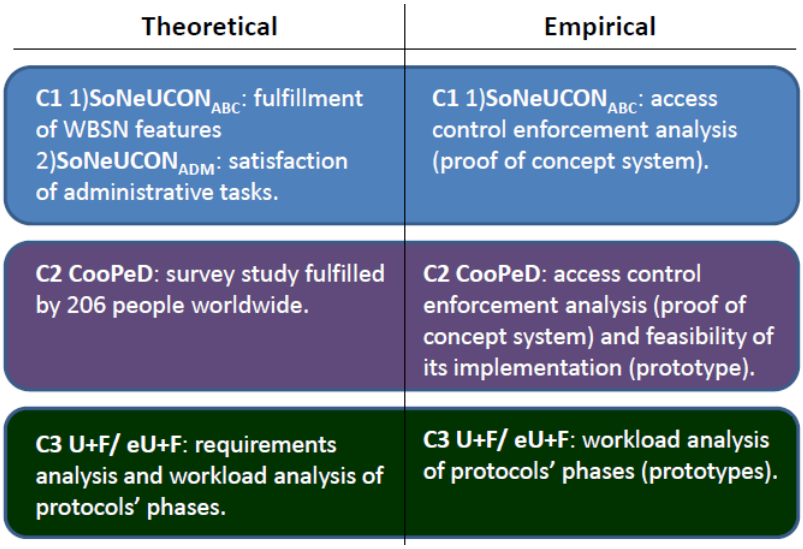


Figure 9.1: Evaluations overview

In sum, $SoNeUCON_{ABC}$ is evaluated in Section 9.1. Subsequently, $SoNeUCON_{ADM}$, the administrative model for $SoNeUCON_{ABC}$, is evaluated in Section 9.2. Section 9.3 presents the evaluation of CooPeD. The evaluation of U+F is shown in Section 9.4. Similarly, the evaluation of eU+F is presented in Section 9.5. Finally, a comparative assessment of U+F and eU+F is described in Section 9.6.

9.1 Evaluation of SoNeUCON$_{ABC}$

$SoNeUCON_{ABC}$ evaluation is performed at two levels, the theoretical (Section 9.1.1) and the empirical one (Section 9.1.2).

9.1.1 Theoretical evaluation

This Section presents the theoretical evaluation. Firstly, if the proposed model manages access control according to the set of identified WBSN features (*distance*, *multi-path*, *direction*, *common-contact*, *clique*, *flexible attributes*) is analysed in Chapter 2. Secondly, the expressive power of the proposed policy language is studied (Section 9.1.1.2).

9.1.1.1 Capability of the model to consider WBSN features

Access control policies are specified in regard to a set of attributes of the requesting subject, the requested object and the set of relationships between the administrator and the requester (respectively, ATT(S), ATT(O) and ATT(RT)). Recalling the definition of $SoNeUCON_{ABC}$ and considering that the relationship management is an essential issue in this model, for a particular request of an action r made by a requester over an object o of an administrator a, the set of managed relationships, rt, corresponds to all enriched paths that have as initial node a and terminal node s. Next, if the six WBSN features described in Section 2.4 can be addressed by the $SoNeUCON_{ABC}$ model is studied.

Distance. It is immediate to show that the model can manage policies that consider indirect relationships between s and a, as rt contains, in theory, all the relationships (direct and indirect) between both entities.

Multi-path. Similarly, as rt contains all the relationships between a and s (in the form of enriched paths), the model allows the definition of access control policies that consider multiple paths.

Direction. In the same way, as the enriched paths comprising rt contain all the edges connecting two consecutive nodes in the path, in the forward and backward directions, to define access control policies that consider unidirectional and bidirectional relationships is possible in the model.

Common-contacts. In a WBSN, assuming the existence of two users, a and s and a set V_l of common contacts, the existence of common-contacts between a and s can be considered in two different ways, where "$\cdots$" refers to the existence or not of additional edges:

1 a has a unidirectional and direct relationship with each $v_{l_i} \in V_l$ and each v_{l_i} has a bidirectional relationship with s. This will be represented by the following enriched path: $\{a, (e^k_{a,l_i}, \cdots; \cdots), v_{l_i}, (e_{l_i,s}, \ \cdots; e_{s,l_i}, \cdots), s\}$

2 a and each v_{l_i} have bidirectional relationships with s. Thus, there are enriched

paths between a and s such that $\{a, (e_{a,l_i}, \cdots; e_{l_i,a}, \cdots), v_{l_i}, (e_{l_i,s}, \cdots;\ e_{s,l_i}, \cdots), s\}$

As rt contains all the relationships (enriched paths) between a and s (depth $\leqslant 6$, recall [153]), a policy considering the existence of common contacts can be evaluated within the model.

Clique. A clique in a digraph D (i.e. directed graph) is referred to a complete digraph between a set of C nodes (including a and s). Then, a clique corresponds to the existence of different bidirectional relationships between all nodes involved in it. In particular, assuming that the number of nodes in the clique distinct than a and s is $N = C - 2$, a clique exists if there are $\sum_{K=1}^{N} P(K, N) + 1$ different enriched paths, such that only N distinct nodes plus a and s are involved in those paths and there exists a bidirectional direct relationship of the same type between all these nodes. Note that $P(K, N)$ refers to the number of K-permutations in a set of N elements. Then, in case $C = 2$ the number of paths is 1, in case $C = 3$, the required number of such paths is 2, for $C = 4$, it is 5, and for $C = 5$, 16 paths are required to exist.

Therefore, given rt, the model allows verifying the existence of a clique of N nodes, and then, it may support access control policies that consider cliques defined in such a way.

Flexible elements in access control policies. In this model, $ATT(S)$, $ATT(O)$, $ATT(RT)$, as well as B and C, are used to define access control policies. This issue together with the definition of a policy language with operators that allow the combination of policy elements, facilitates the definition of access control policies with varied elements and thus, the specification of fine-grained preferences.

Note that although the model allows managing access control in regard to this set of features, to supplement the model with an expressive policy language that also supports them is necessary.

9.1.1.2 Expressive power of the policy language

The pursued goal of $SoNeUCON_{ABC}$ is to reach a high level of expressive power, attesting the management of previous features. To do so and compare the expressive power of this model against those analysed in Section 2.5.3, the possibilities of $SoNeUCON_{ABC}$ to define policies presented in Section 2.4 is studied. It is assumed that the granted right over objects entitled "party" is "read" and not a single condition and/ or obligation has to be satisfied:

P1 Access is granted to users who are friends of neighbours of his/ her relatives if the relationship between his/ her relatives and his/ her relatives' neighbours was established before 2,000. (F1 and F6)

This policy corresponds to an indirect relationship composed of three direct forward relationships from a to s which involve the use of the attribute $role$ in each hop and the attribute $creationYear$ in the second hop.

$\rho = (\varnothing;\ (title = party);\ ((((role = relative);\ (role = neighbour \wedge creationYear < 2000); (role = friend))), \varnothing, \varnothing); read; \varnothing;\ \varnothing)$

P2 Access is granted to users who have three friends in common with the administrator of the requested object. (F2)

One possible option of being common-contact (F4) refers to the existence of an enriched path between a and s where the first hop refers to a direct forward relationship from a to one of his contacts (F3-unidirectional) and the second hop refers to a forward and a backward relationship between such contact and s (F3-bidirectional). Besides, as 3 common-contacts are required, a total of 3 analogous enriched paths have to be identified (F2) and thus, ϖ takes value 3.

$\rho = (\varnothing;\ (title = party);\ ((((role = friend);\ (role = friend) \wedge -(role = friend))),\ 3,\ \varnothing);\ read;\ \varnothing;\ \varnothing)$

P3 Access is granted to users who belong to the clique in which two users and the administrator of the requested object are involved, having all of them a

friendship relationship. (F3)

This policy corresponds to the existence of two enriched paths, one between a and s and one that includes a, s and other user (F5). Then, δ is 3 because 3 users are involved in the clique. Additionally, enriched paths are composed of a direct forward friendship relationship (F6) and implicitly, a backward one.

$\rho = (\varnothing;\ (title = party);\ ((((role = friend))),\ \varnothing,\ 3);\ read; \varnothing;\ \varnothing)$

P4 Access is granted to users who are connected to the administrator by two different paths composed of unidirectional relationships oriented from the requester to the administrator. Moreover, relationships involved in all paths have to be highly trusted. (F4 and F6)

This policy refers to the existence of, at least, a pair of paths with a certain kind of constraints regarding the level of trust of the relationship.

$\rho = (\varnothing;\ (title = party);\ ((((trust = high))),\ 2,\ \varnothing);\ read; \varnothing;\ \varnothing)$

P5 Access is granted to users who are friends of the administrator of the requested object, also having a bidirectional relationship with him/ her. (F5)

This policy refers to the specification of a direct forward and a direct backward friendship relationship.

$\rho = (\varnothing;\ (title = party);\ ((((role = friend) \wedge -(role = friend))), \varnothing,\ \varnothing);\ read;\ \varnothing;\ \varnothing)$

P6 Access is granted to users who are friends of the administrator of the requested object. (F5)

This policy only involves the specification of the role of the relationship.

$\rho = (\varnothing;\ (title = party);\ (((role = friend)));\ read;\ \varnothing;\ \varnothing)$

P7 Access is granted to users if they are females under 30 years old or if they are females under 40 who have studied computer science or if they are females who have studied computer science and physics. (F6)

Table 9.1: WBSNs structure

WBSNs id	$\#e_i$	$\#v_i$	$\overline{e_i/v_i}$
1	2,980,388	50,000	60
2	5,965,777	50,000	120
3	8,949,375	50,000	185
4	10,929,713	50,000	219

ρ = (((*gender* = *female*) ∧ ((*age* < 30) ∨ ((*age* < 40) ∧ (*studies* = *c.science*)) ∨ ((*studies* = *c.science*) ∧ (*studies* = *physics*)))); (*title* = *party*); ∅; *read*; ∅; ∅)

All policies are satisfactorily expressed by $SoNeUCON_{ABC}$. Thus, the suitability of the model for the WBSN field is recognized.

9.1.2 Empirical evaluation

The feasibility of implementing $SoNeUCON_{ABC}$ is analysed studying the Temporal Workload (TW) of policy enforcement. This is performed through the development of a proof of concept system. Firstly, four WBSNs are randomly constructed. Table 9.1 depicts the number of nodes ($\#v_i$), the number of relationships ($\#e_i$) and the mean number of relationships per node ($\overline{e_i/v_i}$) that each WBSN involves. Then, based on developed WBSNs, policy enforcement is studied. It is assumed that the number of hops between a pair of WBSN users is limited to 6 due to theoretical studies [153].

The experimental study of policy enforcement is divided in two steps:

1. *Analysis of rt construction*: For each WBSN, a total of 7 *rt* structures are constructed choosing random requesters and administrators. Table 9.2 details the number of relationships ($\#e_i$ *explo.*) and nodes explored ($\#v_i$ *explo.*) for constructing *rt*, the number of relationships (*rt* $\#e_i$) and nodes (*rt* $\#v_i$) that each final *rt* comprises and the TW of constructing *rt* (*rt* $TW(ms)$). Note that even all *rt* are constructed choosing random users, the amount of nodes and relationships involved in them are considered sufficient to guarantee the appropriateness of the evaluation process.

2. *Policy evaluation*: In each constructed rt, policies proposed in Section 9.1.1.2 are independently evaluated. The TW of performing policy evaluation is summarized in Table 9.3.

Concerning technical details, the proof of concept system is developed in Java 1.7, using a MySQL 5.2 database to store nodes and relationships. Moreover, experiments have been executed over a Pentium D 2.3 GHz with a Lion 10.8 operating system using 500 MB of RAM.

Finally, it should be noticed that the application of graph structures called small-world networks [170] has been considered. This type of graphs are characterized by the fact that most nodes are not neighbours of each other but they can be reached by any node in a small number of hops. Studies have experimented on actual email contacts within an organization or a student social networking site [171], or have crawled Twitter site to work over the real structure [172]. Nonetheless, given the expressive power of $SoNeUCON_{ABC}$, policy enforcement requires managing features not directly considered within small-world networks and current WBSNs. In particular, to manage (F3) *direction* together with (F4) *multi-path*, the existence of multiple unidirectional and bidirectional relationships between pairs of directly connected nodes is essential. Besides, to manage (F6) *flexible elements in access control policies*, the involvement of $ATT(S)$ and $ATT(E)$ in the graph creation process is also indispensable. In sum, random networks have been chosen instead of small-world networks because not all identified WBSN features can be managed without changing the structure of small-world networks. In particular, four WBSNs have been randomly constructed in such a way that nodes are connected through multiple input and output unidirectional relationships and each node and relationship has multiple $ATT(S)$ and $ATT(E)$ attached to it.

9.1.2.1 Analysis of rt construction

The TW of building rt increases exponentially according to the number of explored nodes, that is, rt is built by visiting, recursively, all contacts of each user (starting from the administrator) until the requester is reached (or the maximum path length is reached). Consequently, the TW of constructing rt increases according to the sum of all visited users at each path length, that is, $\sum_{i=1}^{K}(\eta^i)$ where η refers to the average number of users' contacts and K corresponds to the path length. Table 9.2 depicts the TW of constructing multiple rt in each proposed WBSN. In the worst analysed situation, $rt\ id$ = 22, the TW exploring 10,929,713 relationships is 105,478 ms. By contrast, in a better situation, for example, in $rt\ id$ = 12, the TW exploring 119 relationships is 60 ms.

Nonetheless, it should be noticed that, under certain circumstances, some rt involve more nodes and relationships than those that generally appear in a real scenario. Taking Facebook as a representative WBSN, assuming that the average number of Facebook contacts is 190 [173] and the maximum number of hops are two (friend-of-a-friend), the average maximum number of relationships and nodes among a pair of users is $190+190^2 = 36{,}290$. Consequently, rt whose creation involves the exploration of more than 36,290 nodes exceed the average case.

9.1.2.2 Policy evaluation

Proposed access control policies are evaluated and Table 9.3 depicts the TW of their evaluation. All access control policies, except for P3 that refers to cliques construction, are quickly evaluated reaching a TW lower than 90 ms. The most significant rt to analyse, with the highest number of relationships and nodes, are $rt\ id$ = 1, 8, 15 and 22. They involve 83, 164, 502 and 751 relationships and 49, 80, 170 and 251 nodes respectively. Policy evaluation concerning these rt does not exceed 100 ms but for P3. Evaluating policy P3 takes 8,489 ms in $rt\ id$ = 1, more than 100,000 ms in $rt\ id$ = 11 and more than 200,000 ms in $rt\ id$ = 15 and $rt\ id$ = 22.

Table 9.2: Analysis of rt construction

WBSN id = 1					
rt id	$\#e_i$ explo.	$\#v_i$ explo.	$\#e_i$ *rt*	$\#v_i$ *rt*	*rt* TW (ms)
1	252,691	505,382	36	24	4,142
2	3,662	7,324	4	4	435
3	81	162	1	2	28
4	79	158	2	2	54
5	69	120	1	2	38
6	65	130	1	2	51
7	1	2	1	2	13
WBSN id = 2					
rt id	$\#e_i$ explo.	$\#v_i$ explo.	$\#e_i$ rt	$\#v_i$ rt	rt TW (ms)
8	1,958,163	3,916,326	164	80	21,287
9	13,557	27,114	6	5	712
10	139	278	1	2	57
11	126	252	2	2	88
12	119	238	1	2	60
13	115	230	1	2	62
14	1	2	1	2	30
WBSN id = 3					
rt id	$\#e_i$ explo.	$\#v_i$ explo.	$\#e_i$ rt	$\#v_i$ rt	rt TW (ms)
15	6,163,496	12,326,996	502	170	56,811
16	29,771	49,542	6	5	273
17	201	402	1	2	80
18	187	374	2	2	110
19	174	348	1	2	88
20	174	348	1	2	86
21	37	74	1	2	36
WBSN id = 4					
rt id	$\#e_i$ explo.	$\#v_i$ explo.	$\#e_i$ rt	$\#v_i$ rt	rt TW (ms)
22	10,929,713	22,231,690	751	251	105,478
23	445,839	91,678	8	6	1,721
24	244	488	1	2	44
25	230	460	2	2	134
26	216	432	1	2	96
27	216	432	1	2	83
28	33	66	1	2	46

In sum, policy evaluation TW is satisfactory and just cliques management has to be discussed. The attained results may be justified by the fact that the implemented algorithm for searching cliques is not efficient enough and by the fact that experiments are executed in a computer with limited resources.

9.1.2.3 Summary: policy enforcement

Table 9.4 presents the TW of the policy enforcement process, that is, the sum between steps *Analysis of rt construction* and *Policy evaluation*. The tolerable waiting time of WBSN users for information retrieval is approximately 2,000 ms [174]. Thus, results of the implemented proof of concept system are successful in

Table 9.3: Policies evaluation temporal workload

WBSN id = 1							
rt id	P1-TW (ms)	P2-TW (ms)	P3-TW (ms)	P4-TW (ms)	P5-TW (ms)	P6-TW (ms)	P7-TW (ms)
1	1	1	8,489	1	1	<1	<1
2	<1	<1	717	<1	<1	<1	<1
3	<1	<1	247	<1	<1	<1	<1
4	<1	1	753	<1	<1	<1	<1
5	<1	<1	559	<1	<1	<1	<1
6	<1	<1	571	<1	<1	<1	<1
7	<1	<1	463	<1	<1	<1	<1
WBSN id = 2							
rt id	P1-TW (ms)	P2-TW (ms)	P3-TW (ms)	P4-TW (ms)	P5-TW (ms)	P6-TW (ms)	P7-TW (ms)
8	4	17	>100,000	4	2	2	<1
9	<1	<1	1,786	<1	1	<1	<1
10	1	<1	355	<1	<1	1	<1
11	<1	<1	1,138	1	1	<1	<1
12	1	<1	822	<1	<1	1	<1
13	<1	<1	843	1	<1	<1	<1
14	1	<1	729	<1	<1	<1	<1
WBSN id = 3							
rt id	P1-TW (ms)	P2-TW (ms)	P3-TW (ms)	P4-TW (ms)	P5-TW (ms)	P6-TW (ms)	P7-TW (ms)
15	14	5	>200,000	4	2	9	<1
16	1	<1	3,026	1	<1	<1	<1
17	<1	<1	634	<1	1	<1	<1
18	<1	1	1,634	<1	<1	<1	<1
19	1	<1	1,067	1	<1	<1	<1
20	<1	<1	1,014	<1	<1	1	<1
21	<1	1	959	<1	1	<1	<1
WBSN id = 4							
rt id	P1-TW (ms)	P2-TW (ms)	P3-TW (ms)	P4-TW (ms)	P5-TW (ms)	P6-TW (ms)	P7-TW (ms)
22	71	67	>200,000	76	80	18	85
23	<1	1	9,076	<1	<1	<1	1
24	<1	1	1,952	<1	<1	<1	1
25	1	<1	3,683	<1	<1	<1	1
26	<1	1	2,198	<1	<1	<1	1
27	<1	<1	2,149	<1	<1	<1	1
28	<1	<1	1,396	<1	<1	<1	1

most cases. Particularly, they are satisfactory enforcing policies without cliques, if explored nodes do not exceed about 200,000 and 200 relationships per node. Besides, the enforcement of policies with cliques remains successful if less than about 30,000 nodes and 200 relationships per node are explored.

Concerning $rt\ id = 1,\ 8,\ 15,\ 16,\ 22,\ 23$ and 25, that exceed 2,000 ms, some points are discussed to justify such results. Firstly, some rt may involve the exploration of more quantity of nodes and relationships than those that, on average, take place in WBSNs like Facebook. Secondly, despite the hard task of cliques evaluation due to the amount of paths to analyse (recall Section 9.1.1.1), the implemented algorithm could be enhanced to increase performance and reduce the TW. Lastly, contrary to the developed proof of concept system, WBSNs like Facebook apply huge and

Table 9.4: Policy enforcement temporal workload

WBSN id = 1							
rt id	P1-TW (ms)	P2-TW (ms)	P3-TW (ms)	P4-TW (ms)	P5-TW (ms)	P6-TW (ms)	P7-TW (ms)
1	4,143	4,144	7,458	4,143	4,143	4,142	4,142
2	435	435	1,152	435	435	435	435
3	28	28	275	28	28	28	28
4	54	55	807	54	54	54	54
5	38	38	597	38	38	38	38
6	51	51	622	51	51	51	51
7	13	13	476	13	13	13	13
WBSN id = 2							
rt id	P1-TW (ms)	P2-TW (ms)	P3-TW (ms)	P4-TW (ms)	P5-TW (ms)	P6-TW (ms)	P7-TW (ms)
8	21,291	21,304	>121,287	21,291	21,289	21,289	21,287
9	712	712	2,498	712	713	712	712
10	58	57	412	57	57	58	57
11	88	88	1,226	89	89	88	88
12	61	60	882	60	60	61	60
13	62	62	905	63	62	62	62
14	31	30	759	30	30	30	30
WBSN id = 3							
rt id	P1-TW (ms)	P2-TW (ms)	P3-TW (ms)	P4-TW (ms)	P5-TW (ms)	P6-TW (ms)	P7-TW (ms)
15	56,825	56,816	>256,811	56,815	56,813	56,820	56,811
16	274	273	3,299	274	273	273	273
17	80	80	714	80	81	80	80
18	110	111	1,744	110	110	110	110
19	89	88	1,155	89	88	88	88
20	86	86	1,100	86	86	87	86
21	36	37	995	36	37	36	36
WBSN id = 4							
rt id	P1-TW (ms)	P2-TW (ms)	P3-TW (ms)	P4-TW (ms)	P5-TW (ms)	P6-TW (ms)	P7-TW (ms)
22	105,549	105,545	>305,478	105,554	105,558	105,496	105,563
23	1,721	1,722	10,797	1,721	1,721	1,721	1,722
24	44	45	1,996	44	44	44	45
25	135	134	3,817	134	134	134	135
26	96	97	2,294	96	96	96	97
27	83	83	2,232	83	83	83	84
28	46	46	1,442	46	46	46	47

powerful servers which facilitate the celerity of the policy enforcement process.

One last remark is that despite the conclusions drawn from the previous study, a lower TW in current WBSNs is expected because they do not support the management of subject, objects and relationships attributes in the access control enforcement process, as well as they do not consider multiple relationships (forward and/ or backward) between pairs of users.

9.2 Evaluation of SoNeUCON$_{ADM}$

This Section presents the evaluation of $SoNeUCON_{ADM}$, the administrative model for $SoNeUCON_{ABC}$. It consists of comparing the proposed model with

the most challenging and related administrative models, RBAC and $UCON_{ABC}$. $SoNeUCON_{ADM}$ is compared with RBAC administrative model, for being one of the most mature administrative models [175, 110], and with $UCON_{ABC}$ administrative capabilities, for being the model that lays the bases on the proposed one [44, 106].

Administrative tasks, identified in Chapter 5, are depicted and compared in Table 9.5, where symbol '-' implies that a particular task is not studied.

Concerning the association of data with preferences and data with data owners, $SoNeUCON_{ADM}$ only requires to associate preferences (access control policies) to data. Policies are mainly defined over subjects, objects and relationships attributes instead of being attached to specific objects. By contrast, $UCON_{ABC}$ and RBAC pose more restrictive and tedious tasks from the users point of view. In $UCON_{ABC}$ owners define assertions to associate subjects with objects, as well as to associate policies (composed of assertions) with objects [106]. However, in RBAC permissions are assigned to roles and to objects and then, roles are assigned to users.

Delegation is also managed in all compared models, being the $SoNeUCON_{ADM}$ proposal the most flexible one. In $SoNeUCON_{ADM}$ delegating R involves the establishment of access control policies according to subjects, objects and relationship attributes. Moreover, the delegation of all AR involves the execution of the operation DELEGATE to guarantee that, from the moment the operation is enforced, the delegated object becomes property of the delegatee without the possibility of undoing the operation. Conversely, delegation in $UCON_{ABC}$ is limited to R. It is based on specifying assertions associated with particular requesters which composed of objects and subjects attributes [106]. On the other hand, RBAC delegates R and AR through the association of roles to users.

Revocation is another compared task. $SoNeUCON_{ADM}$ manages direct and indirect revocation. The former is performed by owners through the change of attributes and access control policies. On the contrary, indirect revocation is ex-

Table 9.5: Administrative tasks comparison

Tasks		**SoNeUCON**_*ADM*	**UCON**_*ABC* [44, 106]	**RBAC** [110]
Entities identification	Creating, updating and deleting access control preferences	Owners.	Owners.	Owners.
	Associating preferences to data	Not required	-	Owners
	Revocation management	Usage reference monitor and owners	-	Owners
	Delegation management	Owners	-	Owners
Management procedures	Association between preferences with data and data with data owners	Policies are exclusively associated to data owners concerning subjects, objects and relationships attributes.	Assertions associate subjects and objects	Permissions are associated with roles and and data and roles with data owners
	Revocation management	Weak revocation is managed. Attributes and access control policies updates.	Weak revocation is managed. Time assigned to access control policies.	Weak and strong revocation are managed. Owners revoke users from roles according to their decisions.
	Delegation management	Delegation of R and all AR is available. Owners establish access control policies and execute the delegation operation for all AR.	Delegation of R. Assertions associated with particular requesters.	Delegation of R and AR is available. Owners assigned users to roles to delegate.

clusively related to attributes updates, being particularly related to attributes involving time restrictions. Nonetheless, as this model only delegates R and all AR, just weak revocation is at stake. Similarly, $UCON_{ABC}$ manages weak revocation assigning time to access control policies. Moreover, though not described in the original model, Z. Zhang *et al.* proposed a general procedure to manage weak and strong revocation in $UCON_{ABC}$ [176]. On the other hand, RBAC provides functions to weakly and strongly revoke users from roles by removing the assignment of users to roles.

In the light of the proposed analysis, $SoNeUCON_{ADM}$ supports all identified tasks. Indeed, $SoNeUCON_{ADM}$ has a significant advantage, that is, preferences (access control policies) are associated to users instead of to objects and the burden of managing at least as many policies as uploaded objects is avoided. Moreover, it is noticeable that $SoNeUCON_{ADM}$ does not manage strong revocation because cascading delegations are not required. In other words, this model focuses on ownership and then, owners should manage access control in regard to data their posses, either being an entire piece of data or, when co-ownership management takes place, a part of it.

9.3 Evaluation of co-ownership management

CooPeD has been evaluated in three different ways. The TW of policy enforcement is estimated, using the proof of concept system (Section 9.3.1) developed for the evaluation of $SoNeUCON_{ABC}$, to determine the possibility of implementing CooPeD over the model. Next, the feasibility of implementing CooPeD is tested by a prototype development (Section 9.3.2). Lastly, the relevance of co-ownership management and the usefulness and appealing of the proposal is analysed through a survey study (Section 9.3.3).

9.3.1 Policy enforcement for co-ownership management

The TW of policy enforcement in CooPeD, according to the extension of $SoNeUCON_{ABC}$, is evaluated applying a proof of concept system. Indeed, the same proof of concept as the one developed for the evaluation of policy enforcement in $SoNeUCON_{ABC}$ is applied (Section 9.1.2).

CooPeD requires the enforcement of as many access control policies as owners and co-owners are attached to a particular object multiplied by the number of policies established by each of them. Nonetheless, for the sake of simplicity and based on current WBSNs where users create a policy to be applied to chosen objects (e.g. a photo is accessible for friends), it is assumed that each owner/ co-owner establishes a single policy. Moreover, in current WBSNs tagging is the only way to manage co-ownership. The amount of tags per object can be compared with the number of existing owner/ co-owners per object and consequently, to the number of policies to evaluate. According to this issue, it is noticed that popular WBSNs like Facebook or Flickr allow 50 and 75 tags respectively per object[1] [2]. Then, the TW of executing the enforcement of 1, 5, 14, 25, 50 and 75 policies is measured.

Recalling policy enforcement in $SoNeUCON_{ABC}$, it is based on the construction of rt, which consists of the set of relationships between the administrator and

[1] https://www.facebook.com/help/217258071632275–50, last access May 2014
[2] http://www.flickr.com/help/tags/, last access May 2014

Table 9.6: Explored nodes and TW of rt construction.

	$\#v_i$ explo.	TW rt construction
Direct relationships		
rt id=3 (WBSN id =1)	126	28
rt id=11(WBSN id=2)	252	88
Average	189	58
Indirect relationships (depth 2)		
rt id=9 (WBSN id =2)	27,114	712
rt id=16 (WBSN id=3)	49,542	273
Average	38,328	492.5

the requester, and the later evaluation of access control policies. First, the TW of constructing rt will be analysed. Considering that current WBSNs allow the establishment of direct and indirect relationships of length 2 and that 190 is the average number of contacts per user, the construction of rt exclusively composed of direct relationships involves 192 (190+2) explored nodes (including the administrator and the requester) and the construction of rt with only indirect relationships of length 2 involves 36,292 ($190+190^2+2$) explored nodes. Therefore, for rt with direct relationships, the TW is calculated based on the average between rt = 3 and rt = 11 because the amount of explored nodes (189) is close to 192 (see Table 9.6). In particular, the TW constructing rt for rt = 3 and rt = 11 is 28 ms and 88 ms respectively. Concerning indirect relationships, the average TW in calculated based on rt = 9 and rt = 16 as explored nodes are 38,328 and close to the average (see Table 9.6). Specifically, the TW constructing rt for rt = 9 and rt = 16 is 712 ms and 273 ms respectively. Note that these rt id are the ones applied in the evaluation of $SoNeUCON_{ABC}$, Section 9.1.2.

Co-ownership management involves the creation of an rt per owner/ co-owner and the later evaluation of the policy established by each of them. Once the TW of constructing rt has been estimated, the policies evaluation will be also considered. The TW of evaluating 1, 5, 14, 25, 50 and 75 policies per object, can be estimated multiplying the TW of the enforcement of one policy by the amount of policies to evaluate. For instance, given that 712 ms is the TW of carrying out the enforcement of P1 when rt id = 9 (see Table 9.4), the TW of executing the enforcement of 5 policies P1 is estimated as $5 \cdot 712$ ms. Note that, herein, average cases in terms of

rt with direct and indirect relationships are studied and then, the TW constructing *rt* is considered the same one for all pairs of users (owners/co-owners - requesters).

The estimation of the the enforcement process TW for *rt* with direct (*rt* = 3 and *rt* = 11) and indirect (*rt* = 9 and *rt* = 16) relationships is depicted in Table 9.7. Again, based on the fact that the tolerable waiting time of WBSN users for information retrieval is approximately 2,000 ms, in regard to *rt* with direct relationships and excluding P3 (which defines a clique), 34 policies of the same type (P1, P2, P4, P5, P6 or P7) can be evaluated per object without exceeding this limit, that is 58·34 for sets of policies P1, P2, P6 or P7 ms and 58.5·34 ms for sets of policies P4 and P5. On the contrary, when *rt* with indirect relationships are applied, 4 policies of the same type can be evaluated per object, except for P3, without exceeding 2,000 ms, that is 493·4 ms for sets of policies P1, P4 or P5 and 492.5·4 ms for sets of policies P2, P6 or P7. Moreover, regarding P3, in what concerns *rt* with direct relationships a pair of policies can be evaluated without exceeding 2,000, that is 750.5·2. Unfortunately, just the enforcement of a single policy P3 takes 2,898.5 ms.

Along this Section the TW applied in the enforcement of policies of the same type has been estimated. However, another issue to study is the estimation of the TW for set of policies of different types, thus concluding the maximum amount of policies of different types that can be enforced per object request. Depicted in Table 9.8, TW of evaluating sets of different types of policies for *rt* with direct and indirect relationships do not differ from the evaluation of sets of policies of the same type. Therefore, it is estimated that, on average, the enforcement of a policy P1-P7 distinct from P3 takes 58.17 ms for *rt* with direct relationships amd the enforcement of 34 policies of any type (no P3) without exceeding 2,000 ms is possible. For *rt* with indirect relationships, the enforcement of a policy P1-P7 distinct from P3 takes 492.75 ms and then, 4 policies can be evaluated. On the other hand, when cliques management comes into play, that is, the evaluation of

Table 9.7: Average TW policy enforcement for co-ownership management. Analogous types of policies.

rt **composed of direct relationships**						
rt id	1 policy	5 policies	14 policies	25 policies	50 policies	75 policies
P1/ P2/ P6/ P7-TW (ms)						
rt id=3	28	140	392	700	1,400	2,100
rt id=11	88	440	1,232	2,200	4,400	6,600
Average	58	290	812	1,450	2,900	4,350
P3-TW (ms)						
rt id=3	275	1,375	3,850	6,875	13,750	20,625
rt id=11	1,226	6,130	17,164	30,650	61,300	91,950
Average	750.5	3,752.5	10,507	18,762.5	37,525	56,287.5
P4/ P5-TW (ms)						
rt id=3	28	140	392	700	1,400	2,100
rt id=11	89	445	1,246	2,225	4,450	6,675
Average	58.5	292.5	819	1,462.5	2,925	4,387.5
rt **composed of indirect relationships (depth 2)**						
rt id	1 policy	5 policies	14 policies	25 policies	50 policies	75 policies
P1/ P4/ P5-TW (ms)						
rt id=9	712	3,560	9,968	17,800	35,600	53,400
rt id=16	274	1,370	3,836	6,850	13,700	20,550
Average	493	2,465	6,902	12,325	24,650	36,975
P3-TW (ms)						
rt id=9	2,498	12,490	34,972	62,450	124,900	187,350
rt id=16	3,299	16,495	46,186	82,475	164,950	247,425
Average	2,898.5	14,492.5	40,579	72,462.5	144,925	217,387.5
P2/ P6/ P7-TW (ms)						
rt id=9	712	3,560	9,968	17,800	35,600	53,400
rt id=16	273	1,365	3,822	6,825	13,650	20,475
Average	492.5	2,462.5	6,895	12,312.5	24,625	36,937.5

Table 9.8: Average TW policy enforcement for co-ownership management. Different types of policies.

rt **composed of direct relationships**					
No matter type (P1-P7, no P3) - Avg. TW(ms)					
1 policy	5 policies	14 policies	25 policies	50 policies	75 policies
58.17	290.83	814.33	1,454.17	2,908.33	4,362.5
P3 + other types - Avg. TW(ms)					
1 policy P3 + 21 of others	2 policy P3 + 8 of others	3 policy P3			
1,972	1,966.33	2,251.5			
rt **composed of indirect relationships (depth 2)**					
No matter type (P1-P7, no P3) - Avg. TW(ms)					
1 policy	5 policies	14 policies	25 policies	50 policies	75 policies
492.75	2,463.75	6,898.5	12,318.75	24,637.5	36,956.25
P3 - TW(ms)					
1 policy					
2,898.5					

P3, the maximum waiting time for information retrieval is not exceeded for *rt* with direct relationships applying a policy P3 and 21 policies of other types, as well as applying a pair of policies P3 and 8 policies of other types.

In sum, as the evaluation of $SoNeUCON_{ABC}$ points out, cliques management

involves a great amount of TW (Section 9.1.2) and the evaluation of P3 (that considers a clique) exceeds 2,000 ms in the majority of cases. By contrast, the maximum amount of different policies that can be evaluated (excluding P3) are 34 for rt with direct relationships and 4 for rt with indirect relationships. Concerning current WBSNs where users create a policy to be applied to chosen objects and being 14 the average number of tags that users establish per object[3] (analogous to the number of policies to evaluate), the enforcement of 34 policies for rt with direct relationships is considered satisfactory in an average situation. However, the enforcement process for rt with indirect relationships should be enhanced. In any case, in a real world setting, service providers like Facebook or Flickr apply powerful hardware and software mechanisms that help to speed up the access control enforcement process. Besides, pointed out in Section 9.1.2.3, results are expected to be better in current WBSNs because access control enforcement would not involve such amount of assorted elements, eg. multiple relationships between pairs of users.

9.3.2 CooPeD prototype

A prototype to prove the feasibility of implementing CooPeD has been developed in C#, applying a MySQL DB and Emuge CV 2.2.1 to facial recognition. It consists of a web application that allows co-ownership management of photos of people (photos of cars, animals, etc. are a matter of future work). It is expected that the prototype could be linked to a popular WBSN like Facebook in the future. However, given the limitations of the Facebook's API just the Facebook authentication process and photos stored in Facebook are applied in this prototype. Therefore, the use of Facebook simplifies users' authentication management and avoids the storage of photos in an additional DB.

After being authenticated by Facebook, four functions are provided. First, users have to detail some personal attributes and create relationships with other CooPeD users. Second, based on Section 6.2.2.1, owners and co-owners establish

[3] http://www.flickr.com/photos/mariannabolognesi/7073104431/, last access May 2014

access control policies to delegate R. Third, concerning photos (objects) stored in Facebook users can tagged users in them. Each object o_i is decomposed in o_i^j objects. A facial recognition system identifies faces in o_i, thereby recognizing o_i^j. Then, the owner associates each o_i^j with the appropriate user. Assignations consist of tagging users in their o_i^j, becoming these users co-owners of o_i^j. Note that tagging is a form of delegation where owners execute the delegation operation (DELEGATE(v_k,v_j,o,λ)), being v_k the owner, v_j the chosen user, and λ the value AR. Lastly, based on Section 6.2.2.2, users request access to photos. It should be highlighted that policies enforcement involves processing photos to enable only access to the allowed o_i^j and to restrict access to the remaining ones. This is performed by applying hidden techniques such that each o_i^j is covered with opaque, noise or pixelated rectangles from the top to the bottom of the processed o_i, using established tags as reference points.

The prototype architecture, depicted in Figure 9.2, consists of the following components:

- *Data bases (DB)*: a pair of them is distinguished. *FB data DB*, refers to the Facebook DB to authenticate users and manage photos. Additionally, a *Policies&objects parts DB* stores policies, as well as the identities of owners, co-owners and the object parts assigned to each uploaded photo.

- *Management module*: it performs administrative operations. Based on Facebook, users log into the application (*DB authentication module*). Then, a set of tools allow users to create, upload and delete access control policies (*Policies module*) and other sets of tools allows users to upload objects recognizing objects parts and linking them to appropriate users (*Objects module*).

- *Reference monitor*: it verifies access control policies and delivers (if required) the requested object to the *Management module* to be appropriately processed.

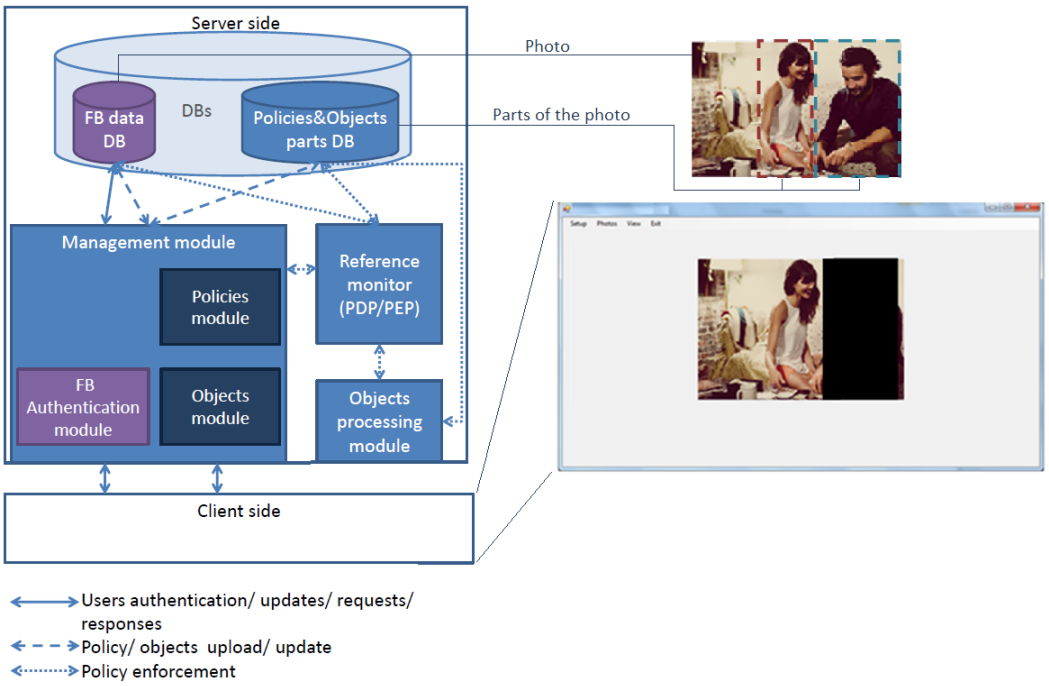


Figure 9.2: CooPeD prototype architecture

- *Objects processing module*: it provides tools to hide parts of photos. After the *Reference monitor* informs about results of policy evaluation, *FB data DB* and *Policies&objects parts DB* provide the requested object and its parts respectively. Next, the object is processed and sent to the *Reference monitor*.

9.3.3 Survey study

A survey is performed to analyse the relevance of co-ownership management and the usefulness and appealing of the proposal. The following sections present the applied methodology (Section 9.3.3.1) and achieved results (Section 9.3.3.2).

9.3.3.1 Methodology

First of all, the goal of the survey is determining the usefulness of CooPeD, highlighting the circumstances under which its use would be desirable.

According to this goal, a total of 9 questions were elicited (Q1-Q9), being all of

them pointed out in Figures 9.3-9.4. Note that in current WBSNs, tagging is the only functionality related to co-ownership management. This is the reason why all defined questions are mainly focused on the use of tags as a means of identifying co-owners.

Afterwards, the survey, which consists of a brief introduction to CooPeD and the proposed questions, was created in Google Drive[4].

In last place, a crawler was developed to send the survey URL worldwide. This program was run for 10 days. After three weeks since the crawler had stopped, 206 people have completed the survey. This amount of people was considered significant and results were gathered.

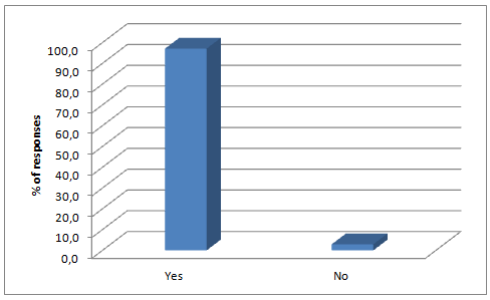
9.3.3.2 Results of the study

Figures 9.3 and 9.4 depict results of the analysis in respect to each question individually. Firstly, regarding Q1, 97.1% of respondents are WBSN users.

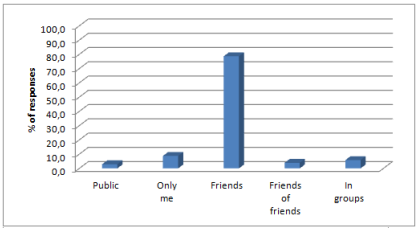
Secondly, the profile of potential users are analysed in Q2-Q5. From Q2 it is highlighted that the majority of respondents, 78.6%, grant access to their data to friends. Besides, in relation to Q3, 49% of the respondents affirm that they have less than 100 photos in their profile and 45.1 % point out that they are tagged in a reduced set of photos. Furthermore, concerning Q4, 45.1% of respondents have few photos in which they are tagged, 29.6% are tagged in most of the photos, 23.3% in about a half and 1.9% in all of them. However, as the plot associated with Q5 depicts, 81.1% of respondents are worried about photos in which they appear but do not control.

Thirdly, the users expected satisfaction is studied regarding Q6-Q9. Results from Q6 point out that a 52.4% of respondents have photos that would not like to be entirely visualized by a person or a group of people. Moreover, concerning Q7, 81.6% of respondents agree with allowing that different users visualize the same photo differently. Specifically, based on the analysis of Q8, 63.6% of respondents

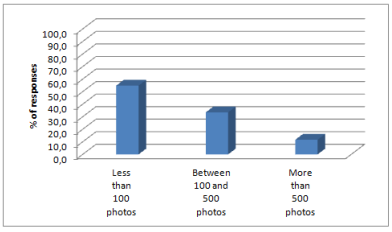
[4]http://www.google.com/drive/, last access May 2014



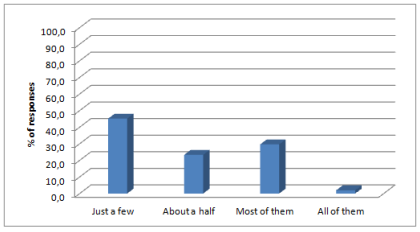
(a) Q1.Are you a user a social network?



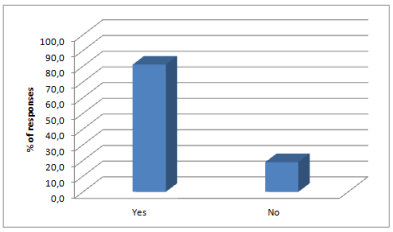
(b) Q2.What privacy preferences do you generally specify in social networks?



(c) Q3.Estimate the number of photos you have on your profile



(d) Q4.In how many photos in which you appear (either uploaded by yourself of by one of your contacts) are you labelled?

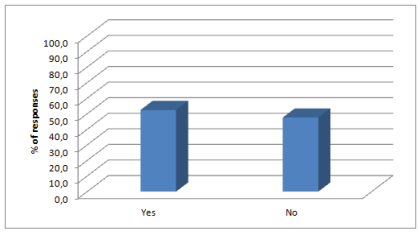


(e) Q5. Are you worried about what might happen with photos in which you are tagged even not being their owner?

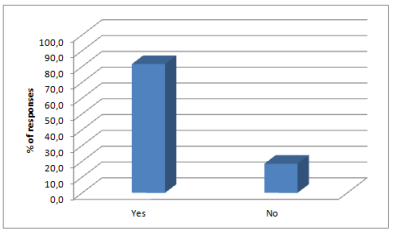
Figure 9.3: Survey study (I)

affirm that they would use the system, 10.2% that would not, 23.8% that only for relevant photos and 2.4% in other cases. Furthermore, the plot related to Q9 shows that 79.1% of respondents affirm that their interest in using CooPeD would increase if sophisticated hidden techniques (e.g. replacement) were applied.

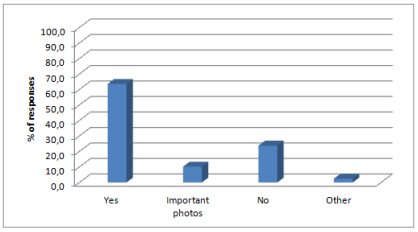
A deep analysis of the profile of respondents, who are WBSN users, is depicted in Table 9.9. In general, respondents choose "Friends" as privacy preferences (Q2) no matter the amount of photos they have in their profiles (Q3). In particular, 75.47% of respondents who have less than 100 photos in their profiles, 86.76% of respondents



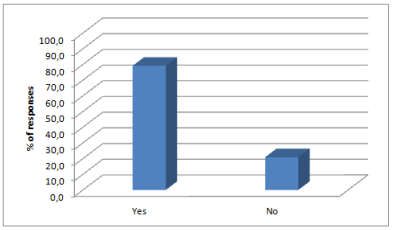
(a) Q6. Do you have photos you would not want someone (or group of people) to see them completely?



(b) Q7. As a photo owner, would you agree with the fact that different users visualize the same photo in a different way?



(c) Q8. Would you use the proposed system?



(d) Q9. Would your interest in using the system increase if the hiding technique was sophisticated?

Figure 9.4: Survey study (II)

who have between 100 and 500 photos in their profiles and 75% of respondents who have more than 500 photos in their profiles, have established "Friends" as privacy preferences. Moreover, regardless of the chosen privacy preferences (Q2) or the amount photos users have in their profiles (Q3), respondents are worried about photos in which they are tagged (Q5). Indeed, all the respondents who have established privacy preferences as "Public", having less than 100 photos and having between 100 and 500 photos in their profiles, are worried about being tagged. Therefore, it can be concluded that co-ownership access control management is worth studying regardless of the kinds of users.

On the other hand, it should be recalled that CooPeD focuses on processing decomposable objects according to owners and co-owners privacy preferences granting access to the appropriate objects' parts. Thus, different users can visualize the same object in a different way. In this regard, Table 9.10 analyses the amount of respondents who, being WBSN users, are potential users of CooPeD. 85.40%

Table 9.9: Analysis of users' profiles, relative percentages

Q3		Q2		Q5	
Estimate the number of photos you have on your profile		What privacy preferences do you generally specify in social networks?		Are you worried about what might happen with photos in which you are tagged even not being their owner?	
Answer	# responses (%)	Answer	# responses (%)	Answer	# responses (%)
Less than 100	106 (53.27%)	Public	3 (2.83%)	Yes	3 (100%)
				No	0 (0%)
		Only me	15 (14.15%)	Yes	13 (86.67%)
				No	2 (13.33%)
		Friends	80 (75.47%)	Yes	65 (81.25%)
				No	15 (18.75%)
		Friends of friends	2 (1.89%)	Yes	1 (50%)
				No	1 (50%)
		Groups	6 (5.66%)	Yes	6 (100%)
				No	0 (0%)
Between 100 and 500	69 (34.67%)	Public	2 (2.90%)	Yes	0 (0%)
				No	2 (100.0%)
		Only me	0 (0%)	Yes	0 (0%)
				No	0 (0%)
		Friends	60 (86.76%)	Yes	54 (90.00%)
				No	6 (10.00%)
		Friends of friends	5 (7.25%)	Yes	1 (20.00%)
				No	4 (80.00%)
		Groups	2 (2.90%)	Yes	1 (50.00%)
				No	1 (50.00%)
More than 500	24 (12.06%)	Public	1 (4.17%)	Yes	1 (100.0%)
				No	0 (0%)
		Only me	0 (0%)	Yes	0 (0%)
				No	0 (0%)
		Friends	18 (75%)	Yes	12 (66.67%)
				No	6 (33.33%)
		Friends of friends	1 (4.17%)	Yes	0 (0%)
				No	1 (100.0%)
		Groups	4 (16.67%)	Yes	3 (75.00%)
				No	1 (25.00%)

of respondents who have photos which they want to show partially (Q6:yes), allow different visualizations of a photo by different users (Q7:yes). Likewise, from the set of respondents who do not have photos to hide partially (Q6:no), 78.35% of them also allow different users to visualize the same photo in a different way (Q7:yes).

Additionally, the analysis of users who would use the system (Q8) is essential to identify potential users. From the set of respondents who accept different visualization of a photo (Q7:yes) and have photos to disclose partially (Q6:yes), 86.68% of them would use the system in any case and 18.18% for relevant photos. Furthermore, from the set of respondents that allow a different visualization of a photo (Q7:yes) and assuming that respondents who do not currently have photos

Table 9.10: Analysis of potential users, relative percentages

Q6		Q7		Q8		Q9	
Do you have photos you would not want someone (or group of people) to see them completely?		As a photo owner, would you agree with the fact that different users visualize the same photo in a different way?		Would you use the proposed system?		Would your interest in using the system increase if the hiding technique was sophisticated?	
Answer	# responses (%)	Answer	# responses (%)	Answer	# responses (%)	Answer	# responses (%)
Yes	103 (51.50%)	Yes	88 (85.40%)	Yes	71 (80.68%)*	Yes	63 (88.73%)
						No	8 (11.27%)
				No	1 (1.14%)	Yes	1 (100.0%)
						No	0 (0%)
				Only relevant	16 (18.18%)*	Yes	14 (87.5%)
						No	2 (12.5%)
		No	15 (14.56%)	Yes	9 (60.00%)	Yes	7 (77.78%)
						No	2 (22.22%)
				No	3 (20.00%)	Yes	2 (66.67%)
						No	1 (33.33%)
				Only relevant	3 (20.00%)	Yes	2 (66.67%)
						No	1 (33.33%)
No	97 (48.50%)	Yes	76 (78.35%)	Yes	44 (57.89%)*	Yes	37 (84.09%)
						No	7 (15.91%)
				No	6 (7.89%)	Yes	3 (50.0%)
						No	3 (50.0%)
				Only relevant	26 (34.21%)*	Yes	21 (80.77%)
						No	5 (19.23%)
		No	21 (21.65%)	Yes	7 (33.33%)	Yes	4 (57.14%)
						No	3 (42.86%)
				No	10 (47.62%)	Yes	2 (20.0%)
						No	8 (80.0%)
				Only relevant	4 (19.05%)	Yes	1 (25.0%)
						No	3 (75.0%)

*: potential users.

to disclose partially (Q6:no) they may have in the future, 57.89% of them would use CooPeD in any case and 32.21% for relevant photos. Note that the remaining set of cases (namely, namely, Q6:yes/no followed by Q7:no) are not relevant for the analysis because respondents involved in such sets do not allow a photo to be differently visualized by different users and it is essential to use CooPeD.

In the light of the foregoing results, potential users of CooPeD corresponds to respondents who having or not photos to partially disclose (Q6:yes or no), allow a photo to be differently visualized by different users (Q7:yes) and would use the system in any case (Q8:yes) or for relevant photos (Q8:only relevant). Thus, identified in Table 9.10 with symbol $*$, potential users correspond to 78.5% ($\frac{(71+16+44+26)\cdot 100}{200}$) of the set of respondents who are WBSN users and 76.2% ($\frac{(71+16+44+26)\cdot 100}{206}$) in respect to the total amount of respondents.

As a final remark, from Table 9.10 it is noticed that, in general, the interest of using the system would increase (Q9:yes) in case of applying hiding sophisticated

techniques. More specifically, respondents who would use the system in any case or for relevant photos are the most interested in applying hiding sophisticated techniques. Besides, the interest of respondents who would not use the system would also increase applying such hiding techniques.

9.4 Evaluation of U+F

This Section presents the evaluation of U+F from a theoretical (Section 9.4.1) and an experimental (Section 7.2.2) point of view.

9.4.1 Theoretical evaluation

Looking backward at U+F, the goal of this protocol is the development of an architecture along with a protocol to achieve interoperability and reusability between multiple WBSNs. The theoretical evaluation is carried out discussing the satisfaction of established requirements (Section 9.4.1.1) and analysing the performance of the protocol (Section 9.4.1.2).

9.4.1.1 Requirements evaluation

In this Section the degree of satisfaction of requirements presented in Section 7.2.2 is analysed.

Concerning **interoperability and reusability in regard to direct relationships**, it is achieved due to the decentralization of identity data, resources and access control policies management, being all of them stored in IdPs, Hosts and AMs respectively. Data can be replaced, moved or updated without affecting any service of WBSNs. Moreover, different WBSNs can make use of the same resources, identity data and access control policies if the same IdPs, Host and AMs are linked to them. More specifically, regarding interoperability, the use of the same identity data specification, FOAF files in this case, and the use of a concrete application of UMA, including the specification of claims and the Fat Requester, address this

issue.

The second requirement is **resources and identity data confidentiality and access control**. This protocol is based on UMA and adds a concrete claims management mechanism. In a nutshell, access control focuses on the satisfaction of access control policies after proving the appropriate claims, that include a proof of the existence of a relationship between the administrator of the requested resources or identity data and the requester. Afterwards, a token, with a particular expiration time, is delivered according to the satisfied policies and then, access is granted until the token expires. Indeed, the expiration time is essential to avoid adversaries using tokens some time after their delivery. Nonetheless, tokens expiration time does not have to be too long because if tokens are accessed, adversaries could use them.

Besides, when the token is presented to IdPs or Hosts to get the requested resources or identity data, its verification asserts that the entity which presents the token is the same one to which the token was initially delivered, thereby preventing impersonations of IdPs and Hosts.

In addition, WBSNs may act on behalf of users and get access to data while users are logged. Therefore, to prevent WBSNs from acting on behalf of users when they are not logged, each user authenticates himself against his Host and IdP in the log-in and log-out in the WBSN.

Other requirement is **resources and identity data integrity**. SSL, or other channel with analogous characteristics, is applied in the protocol to guarantee that adversaries do not alter messages content, thus reaching integrity of resources and identity data. Moreover, this channel also prevents adversaries getting access to interchanged data.

The third requirement is **chain of trust**. Users establish is their IdPs, Hosts, AMs and WBSNs the appropriate lists of trusted IdP_CAs, WBSN_CAs and WBSNs. Then, some interchanged messages are signed by issuer entities as well as by passing-through entities to finally verify that signer entities are within stored lists.

Thus, being in a list means being trusted. In particular, messages related to the acquisition of claims are properly signed by IdPs and AMs and messages interchanged between WBSNs are signed by the WBSN at which the user requests access.

As a result, lists of trusted entities and signatures are mechanisms apply to prevent WBSNs and IdPs impersonations. Indeed, it should be noticed that having different certificates for signing (delivered by IdP_CAs and WBSN_CAs) and authenticating (applied in the communication channel using e.g. SSL) helps to make difficult IdPs and WBSNs impersonations. This is comparable with devices like the Spanish national identity card (DNIe) which owns both an authentication and a signature certificate to prevent unauthorized uses.

On the other hand, **access the minimum identity data** is related to claims management. For example, the easiest way would be the interchange of complete FOAF files between WBSNs. Nevertheless, to satisfy the proposed requirement, identity data interchanged between WBSNs is limited to users identifiers and the WBSNs in which users are enrolled in.

9.4.1.2 Performance analysis

Trying to attain more specific results and recalling the U+F access control management is based on *SoNeUCON* model, this Section analyses the performance of U+F execution phases and *SoNeUCON* managed features. In particular, the number of messages exchanged, the number of entities involved, the number of performed signatures and the amount of signatures verifications, are analysed.

Protocol phases analysis Table 9.11 presents the performance analysis per each protocol phase (Ph). Besides, in order to have a general perception of U+F executions, phases related to a user logs in to a WBSN (Ph2) and a user accesses a contact's data (Ph3) are studied regarding the worst and best case. In relation to the worst case, it is assumed that all messages of the protocol are interchanged because no data is stored and reused. On the contrary, according to the best case,

Table 9.11: U+F theoretical evaluation: protocol phases

Phases	Entities	Signatues	Signatures verification	# Messages
(Ph1)Initialization	I+H+A+1	*	*	$12{\cdot}(I+H)+2\cdot A$
(Ph1.1)Entities registration	I+H+A+1	*	*	$10{\cdot}(I+H)+2\cdot A$
(Ph1.2)Registration of resources and identity data	I+H	*	*	$2{\cdot}(I+H)$
(Ph1.3)Specification of information in WBSNs	1+S	*	*	*
Worst case				
(Ph2)Log-in	$6+C_I+C_{WBSN}$	2	2	24
(Ph2.1)Authentication	*	*	*	*
(Ph2.2)FOAF file acquisition	$3+C_I+C_{WBSN}$	1	1	12
(Ph2.3)Resource acquisition	$3+C_I+C_{WBSN}$	1	1	12
(Ph3)Access direct contact	$13+C_I+C_{WBSN}$	12	12	50
(Ph3.1)FOAF file acquisition	$6+C_I+C_{WBSN}$	6	6	25
(Ph3.2)Resource acquisition	$7+C_I+C_{WBSN}$	6	6	25
Best case				
(Ph2)Log-in	$6+C_I+C_{WBSN}$	2	1	17
(Ph2.1)Authentication	*	*	*	*
(Ph2.2)FOAF file acquisition	$3+C_I+C_{WBSN}$	1	1	11
(Ph2.3)Resource acquisition	$3+C_I+C_{WBSN}$	1	0	5
(Ph3)Access direct contact	$13+C_I+C_{WBSN}$	2	4	18
(Ph3.1)FOAF file acquisition	$6+C_I+C_{WBSN}$	1	4	9
(Ph3.2)Resource acquisition	$7+C_I+C_{WBSN}$	1	0	9

N: (# of users in the relationship)-1, N>1
I: # of IdPs of a user
A: # of AMs of a user
*: an element/ action not detailed
H: # of Hosts of a user
C_x: # of IdP_CAs and WBSN_CAs, where x is I or WBSN regarding the type of CA
- : an element/ action not required

it is assumed that tokens are stored and reused.

From Table 9.11 some relevant features can be inferred. Regarding the number of messages exchanged, the quantity of messages in the initialization phase (Ph1) is significant. In particular, the amount of messages increase concerning the number of Hosts, IdPs and AMs used in the registration phase (Ph1.1). Similarly, in the registration of resources and identity data (Ph1.2), messages increase according to the number of used IdPs and Hosts. Nonetheless, the most significant exchange of messages corresponds to the acquisition of resources which, in the worst case, corresponds to 12 in the log-in phase (Ph2.3) and 25 in accessing to a contact's data (Ph3.2). Nonetheless, considering the repeated use of claims and tokens, as it is shown in the best case, the number of messages can be significantly lower, 5 in the log-in phase (Ph2.3) and 9 accessing to a contact's data (Ph3.2).

In what concerns entities, the use of multiple AMs, Hosts and IdPs is specially significant in registration processes (Ph1.1). Nonetheless, the use of a huge quantity

of these entities is not expected. For example, one IdP per user is expected.

Last but not least, signatures are applied in claims authentication. A total of 6 signatures and signatures verifications are performed in the worst case when acquiring a contact's identity data (Ph2.1) and a contact's resource (Ph2.2). By contrast, in the best case, the number of signatures and signatures verifications is significantly reduced.

9.4.2 Experimental evaluation

To empirically analyse the performance of the protocol and evaluate the applicability of U+F, a prototype has been implemented (Section 9.4.2.1) and its temporal workload has been measured (Section 9.4.2.2). Besides, results have been compared with two challenging WBSNs, Facebook and MySpace (Section 9.4.2.3).

9.4.2.1 U+F prototype description

This Section presents the development of a prototype to prove the viability of implementing U+F in a simulated environment. It is composed of two WBSNs, FriendBook+$_{v0}$ and MyLeisure$_{v0}$. The general architecture is depicted in Figure 9.5. A couple of IdPs, a couple of Hosts, a couple of WBSNs and four AMs (one for each Host and IdP) are the entities at stake. Thus, a total of eight servers are used and located in different places along a local network. The key point is to verify that data of MyLeisure$_{v0}$ remains available to FriendBook+$_{v0}$ and the other way round. Users' identity data (profile and contacts) corresponds to their name, nationality, age, email, school and contacts relationships.

Firstly, the existence of a pair of users is assumed, Alice and Bob. Alice is enrolled in FriendBook+$_{v0}$ and Bob is enrolled in MyLeisure$_{v0}$. Then, regarding Figure 9.5, Alice establishes her identity data in IdP1, resources in Host1 and her access control policies in AM_IdP1 and AM_Host1. By contrast, Bob establishes his identity data in IdP2 and resources in Host2 and also uses a couple of AMs,

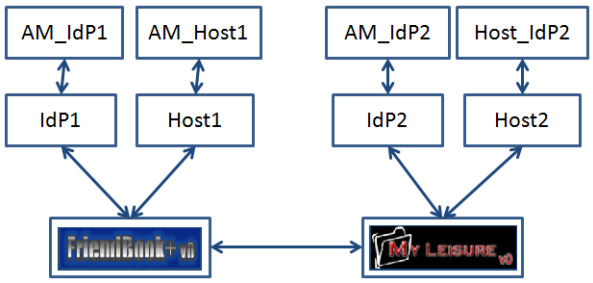


Figure 9.5: Designed U+F prototype architecture

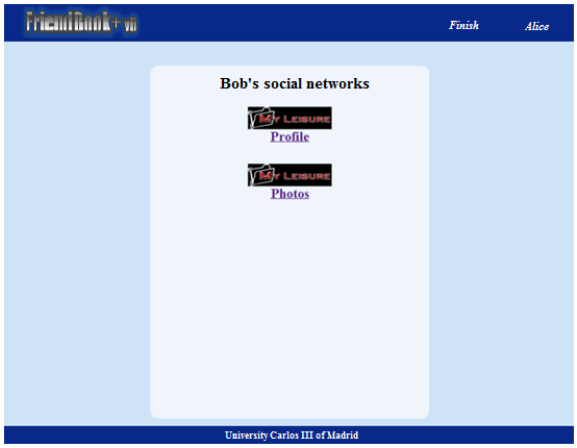


Figure 9.6: Alice access Bob's data

AM_IdP2 and AM_Host2, to establish and manage access control policies. Afterwards, once Alice enrols in FriendBook+$_{v0}$, IdP1 and Host1 are linked to it. Likewise, when Bob enrols in MyLeisure$_{v0}$, he specifies where his identity data and resources are stored, that is, in IdP2 and Host2 respectively. The experimental evaluation verifies that Alice from FriendBook+$_{v0}$ is able to access Bob identity data and resources in MyLeisure$_{v0}$ as Figure 9.6 depicts.

Different technologies have been applied in the prototype development. J2EE and J2SE 1.6 has been used for implementing the pair of proposed WBSNs, FriendBook+$_{v0}$ and MyLeisure$_{v0}$. Glassfish 3.0.1 has been applied to manage IdPs, Hosts and AMs and MySQL 5.2.27 to store data. Additionally, to measure network communications, Firebug[5] 1.7.3 (a Firefox extension) has also been used. On the

[5]http://getfirebug.com/

Table 9.12: Analysing the reuse of data in U+F

		Worst case	Best case	% reuse	Avg. Reuse %		$\mho_{maxReuse}$
# Messages	(Ph1) Log-in	24	17	29,17	46.59	44.12	0,56 (1-44%)
	(Ph2)Access a contact's data	50	18	64			
Signatures	(Ph1) Log-in	2	2	0	41.66		
	(Ph2)Access a contact's data	12	2	83.33			

other hand, photos managed in this implementation have a size between 200kb and 300kb. This size is chosen as an average considering [177]. Moreover, the prototype has been tested in a processor of 3.00GHz (2 CPUs) with 4096MB of RAM.

9.4.2.2 U+F temporal workload

The performance of the protocol is evaluated analysing the TW through the measurement of the cost of accessing to the personal profile and personal photos presented in FriendBook+$_{v0}$ (WBSN1) and the cost of accessing to a contact's profile and photos shown in MyLeisure$_{v0}$ (WBSN2) ($C_{dataAccess}$). Mentioned along the protocol description, some elements (mainly tokens) can be reused and thus, except in the acquisition of personal identity data which is the first data obtained, workload is multiplied by the amount of elements not reused, $\mho$. Depicted in Table 9.12 and based on the worst case (no elements are reused) and the best case (all possible elements are reused) of the amount of performed signatures and interchanged messages when a user logs in to a WBSN (Ph2) and a user accesses a contact's data (Ph3) (recall Table 9.11), the average percentage of elements that can be reused is 44.12% at most, 46.59% regarding interchanged messages and 41.66% in regard to signatures. Due to this analysis, $\mho$ is 1 when elements are not reused, 0.78 if 50% of elements are reused and 0.56 is case of reusing all possible elements, that is 44.12%. As a result, the temporal workload TW is calculated as Equation 9.1 presents.

$$TW = C_{dataAccess} \cdot \mho \tag{9.1}$$

The workload has been analysed computing the average of 10 executions without the reuse of elements, that is, $\mho = 1$. Plot depicted in Figure 9.7 presents the

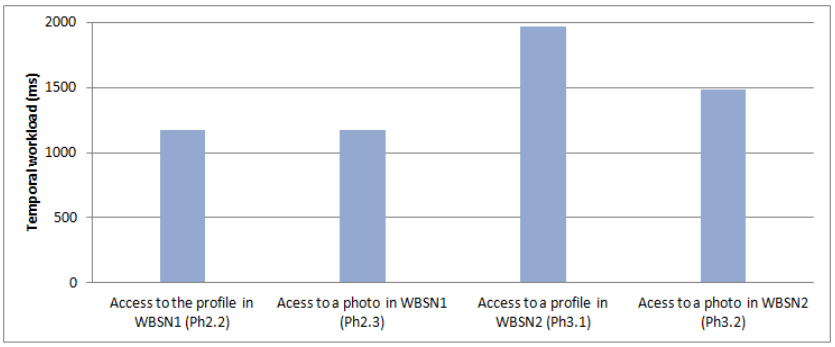


Figure 9.7: U+F temporal workload

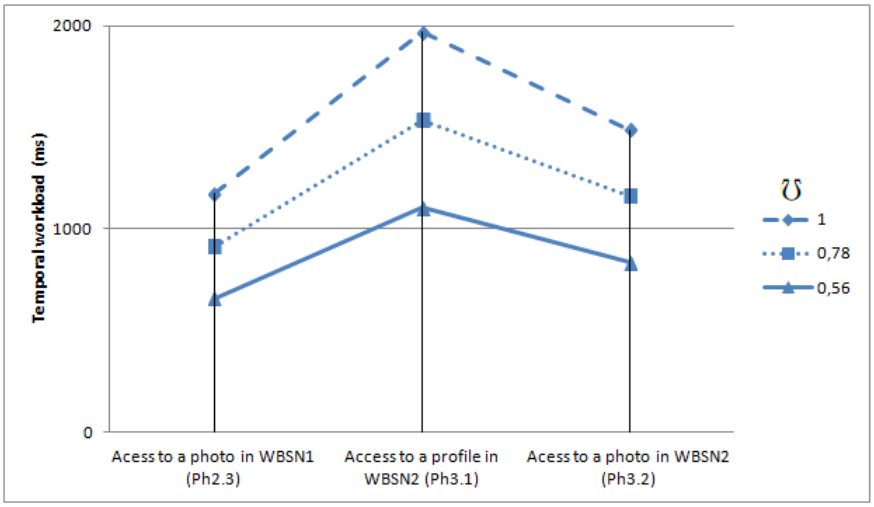


Figure 9.8: U+F temporal workload reused

workload achieved in the protocol's phases (Ph). In particular, 1,176 ms are required to access the profile (FOAF file) in WBSN1 (Ph2.2), 1,174 ms to access to a photo (resource) in WBSN1 (Ph2.3), 1,969 ms to access to a profile in WBSN2 (Ph3.1) and 1,489 ms to access to a photo in WBSN2 (Ph3.2). As expected, it increases when obtaining data of other WBSN, that is, accessing data of a direct contact. The difference of accessing to data in WBSN in relation to WBSN2 is 793 ms for profiles and 315 ms for photos. These results are reasonable because photos are immediately delivered once the access token is presented and in the other case the requested FOAF file has to be processed. In other words, instead of delivering the whole FOAF file, a reduced one is created according to established access control policies.

On the other hand, Figure 9.8 presents the TW according to established values of $\mho$. From the analysis the importance of reusing is noticeable. In particular, if elements are not reused ($\mho$ = 1) 1,173 ms, 915 ms and 657 ms are the TW accessing to a photo in WBSN1 (Ph2.3), to a profile in WBSN2 (Ph3.1) and to a photo in WBSN2 (Ph3.2) respectively. On the contrary, considering an average reuse ($\mho$ = 0.78), the workload is 1,969 ms, 1,536 ms and 1,103 ms and in case of the maximum reuse ($\mho$ = 0.56) the workload is 1,489 ms, 1,161 ms and 834 ms.

9.4.2.3 Temporal workload comparison

U+F has been compared with a pair of currently successful WBSNs, Facebook and MySpace. To perform the comparison a pair of accounts have been created in both WBSNs (two users per WBSN) and they have been connected such that each user has a contact. Moreover, the same set of photos uploaded to FriendBook+$_{v0}$ and MyLeisure$_{v0}$ has been uploaded to the created accounts. Afterwards, using Firebug 1.7.3 (a Firefox extension), the TW of getting the personal profile (obtained when logging) and a personal photo and the workload of getting the profile and a photo of a contact has been measured. Note that although the prototype does not implement the authentication, that is included in the login phase, the prototype workload of accessing to the personal profile of a user can be compared with the one measured in Facebook$_{v0}$ and MySpace$_{v0}$ because authentication techniques of WBSNs like these are based on passwords and thus, the workload of such technique can be disregarded.

Results are depicted in Figure 9.9. The TW regarding the access to the personal profile (Ph2.2), to a personal photo (Ph2.3), to other user's profile (Ph3.1) and to other user's photo (Ph3.2) is, in Facebook, 4,052 ms, 766 ms, 2,556 ms and 624 ms, in MySpace, 4,423 ms, 626 ms, 1,438 ms and 842 ms and, in the prototype, 1,175 ms, 1,173 ms, 1,964 ms and 1,489 ms, respectively. Results show that accessing to the personal profile in MySpace produces the highest workload. This issue is

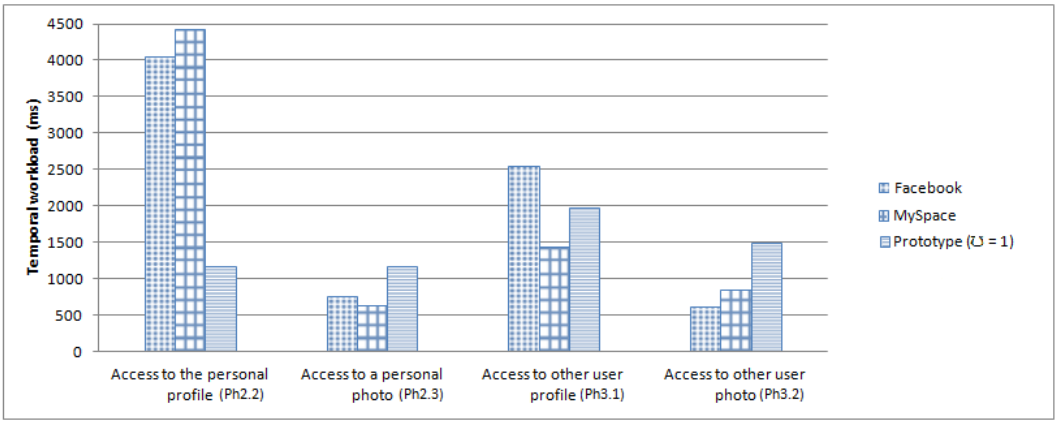


Figure 9.9: Temporal workload comparison

caused by the fact that MySpace profiles include photos and videos. Indeed, it is expected that MySpace or Facebook profiles produce higher workload than the prototype because they are much more richer. By contrast, the prototype produces the highest workload accessing to photos. As expected, the workload is specially significant when accessing data of a user enrolled in other WBSN, differing 547 ms and 647 ms from MySpace and 407 and 865 ms from Facebook when accessing a personal photo(Ph2.3) and a photo of other user (Ph3.2) respectively.

Summing up, there are a couple of points to highlight against and in favour of the developed prototype. On the negative side, in the executions performed for this experimental evaluation, entities (IdPs, Host,...) run in a local network and $C_{dataAccess}$ may be higher in a real environment, e.g. network delays. On the positive side, contrary to the prototype, it is presumable that big companies, which develop WBSNs like Facebook and MySpace and LinkedIn, own robust and efficient infrastructures and mechanisms, for instance cache servers, which help to speed users' requests, thereby achieving successful $C_{dataAccess}$ times. This is not the case of the developed prototype. In conclusion, from the authors' point of view, taking into account the challenge of dealing with indirect relationships in an interoperable environment, the workload of eU+F can be considered reasonable.

9.5 Evaluation of eU+F

Following the same evaluation procedure as for U+F, the extended protocol is evaluated theoretically (Section 9.5.1) and empirically (Section 9.5.2).

9.5.1 Theoretical evaluation

Firstly, the satisfaction of eU+F requirements is analysed (Section 9.5.1.1). Secondly, a performance analysis is carried out (Section 9.5.1.2).

9.5.1.1 Requirements evaluation

eU+F has to fulfil all proposed requirements (Section 8.2.2). In this regard, in the following paragraphs each of them is analysed, identifying their level of satisfaction.

Interoperability and reusability in respect to direct and indirect relationships is the first noticeable requirement. Alike U+F, both features are achieved decentralizing identity data, resources and access control policies management. More specifically, indirect relationships are managed through the inclusion of element *depth* within claims and the management of relationship proofs concerning users involved in them.

The following requirement is **data privacy preservation against WBSNs** and it is divided in two. On the one hand, WBSNs can not access users' data because data as well as decryption keys are encrypted and their decryption is performed at users' browsers, achieving data exposure minimization. Indeed, resources are stored encrypted and identity data and decryption keys are encrypted once being delivered. Then, even an adversary with a valid access token could get access to data. Similarly, AMs impersonations are avoided by signatures of claims requests, together with the encryption of resources and identity data because, neither IdPs would deliver claims to AMs if claims signature verifications fail, nor a single entity even owning a token deliver by a malicious AM would get access to encrypted resources and identity data.

Furthermore, it should be noticed that the local decryption must be performed under security constraints, that is, decrypted data cannot leave the user's browser. Furthermore, the fact that Hosts store encrypted resources and, analogous to WBSNs, they cannot access data is remarkable.

On the other hand, as in U+F, **access the minimum data** is related to claims management. In case of indirect relationships, data interchanged among WBSNs is restricted to proofs that certify the relationship between the administrator of the requested data and the requester. Therefore, this requirement is satisfied to a very large extent, though leaving as an open issue that WBSNs know the relationships proofs.

To conclude, **simple key management** is the last requirement. The proposed cryptographic schemes suppose the creation of as many asymmetric key pairs and symmetric keys as desired. However, assuming that keys are indispensable in any cryptographic approach, in the proposed schemes they are not interchanged out of band. Then, key management is simplified. Indeed, removing out of band interchanges of keys prevents from possible management confusions either intentionally or not. Besides, considering the large quantity of WBSNs and the amount of established relationships, out of band interchanges may become unmanageable.

9.5.1.2 Performance analysis

This evaluation analyses the performance of every eU+F execution phase, eU+F access control management features according to $SoNeUCON_{ABC}$ model and proposed cryptographic schemes. Specifically, the number of entities involved, the number of encryptions and decryptions carried out, the number of signatures and signatures verification performed and the number of messages interchanged are evaluated.

Protocol phases analysis Table 9.13 analyses each protocol phase (Ph) in the worst and best case, that is when elements are not reused and when all possible

elements are reused. Note that data exposure minimization techniques are studied in the following Section.

Table 9.13: eU+F theoretical evaluation: protocol phases

Phases	Entities	Encryptions	Decryptions	Signatures	Signatures verification	# Messages
(Ph1)Initialization	$2{\cdot}I + 2 \cdot H$+ A+S +2	R	*	*	*	$12{\cdot}(I+H)$+ $2{\cdot}A$
(Ph1.1)Entities registration	I+H+A+ 1	*	*	*	*	$10{\cdot}(I+H)$+ $2{\cdot}A$
(Ph1.2)Registration of resources and identity data	I+H	R	*	*	*	$2{\cdot}(I+H)$
(Ph1.3)Specification of information in WBSNs	1+S	*	*	*	*	*
		Worst case				
(Ph2)Login	6+2 $\cdot C_I$+ 2 $\cdot C_{WBSN}$+ 2 $\cdot C_A$	-	1	4	4	24
(Ph2.1)Authentication	*	*	*	*	*	*
(Ph2.2)FOAF file acquisition	3+C_I+C_{WBSN}+ C_A	-	-	2	2	12
(Ph2.3)Resource acquisition	3+C_I+C_{WBSN}+ C_A	-	1	2	2	12
(Ph3)Access to direct contact	13+2 $\cdot C_I$+ 2 $\cdot C_{WBSN}$+ 2 $\cdot C_A$	4	4	16	16	50
(Ph3.1)FOAF file acquisition	6+C_I+C_{WBSN}+ C_A	2	2	8	8	25
(Ph3.2)Resource acquisition	7+C_I+C_{WBSN}+ C_A	2	2	8	8	25
(Ph4)Access to indirect contact	6 + 7 $\cdot N$ + 2 $\cdot C_I$+ 2 $\cdot C_{WBSN}$+2 $\cdot C_A$	4 $\cdot(N+1)$	4 $\cdot(N+1)$	16 $\cdot(N+1)$	16 $\cdot(N+1)$	50 $\cdot(N+1)$
(Ph4.1)FOAF file acquisition	3 $\cdot(N+1)$ + C_I+ C_{WBSN}+C_A	2 $\cdot(N+1)$	2 $\cdot(N+1)$	8 $\cdot(N+1)$	8 $\cdot(N+1)$	25 $\cdot(N+1)$
(Ph4.2)Resource acquisition	3 + 4 $\cdot N$ + C_I+ C_{WBSN}+C_A	2 $\cdot(N+1)$	2 $\cdot(N+1)$	8 $\cdot(N+1)$	8 $\cdot(N+1)$	25 $\cdot(N+1)$
		Best case				
(Ph2)Login	6+2 $\cdot C_I$+ 2 $\cdot C_{WBSN}$+ 2 $\cdot C_A$	-	1	2	2	18
(Ph2.1)Authentication	*	*	*	*	*	*
(Ph2.2)FOAF file acquisition	3+C_I+C_{WBSN}+ C_A	-	-	2	2	12
(Ph2.3)Resource acquisition	3+C_I+C_{WBSN}+ C_A	-	1	-	-	6
(Ph3)Access to direct contact	13+2 $\cdot C_I$+ 2 $\cdot C_{WBSN}$+2 $\cdot C_A$	4	4	2	2	18
(Ph3.1)FOAF file acquisition	6+C_I+C_{WBSN}+ C_A	2	2	1	1	9
(Ph3.2)Resource acquisition	7+C_I+C_{WBSN}+ C_A	2	2	1	1	9
(Ph4)Access to indirect contact	6 + 7 $\cdot N$ + 2 $\cdot C_I$+ 2 $\cdot C_{WBSN}$+2 $\cdot C_A$	4$\cdot(N+1)$	4$\cdot(N+1)$	2$\cdot(N+1)$	2$\cdot(N+1)$	18 $\cdot(N+1)$
(Ph4.1)FOAF file acquisition	3 $\cdot(N+1)$ +C_I+ C_{WBSN}+C_A	2 $\cdot(N+1)$	2 $\cdot(N+1)$	1 $\cdot(N+1)$	1 $\cdot(N+1)$	9 $\cdot(N+1)$
(Ph4.2)Resource acquisition	3 + 4 $\cdot N$ + C_I+ C_{WBSN}+C_A	2 $\cdot(N+1)$	2 $\cdot(N+1)$	1 $\cdot(N+1)$	1 $\cdot(N+1)$	9 $\cdot(N+1)$

N: relationship length, N>1
I: # of IdPs of a user
A: # of AMs of a user
∗: an element/ action not detailed
C_x: # of IdP_CAs, AM_CAs and WBSN_CAs, where x is I, WBSN
H: # of Hosts of a user
R: # of resources of a user
or A regarding the type of CA
- : an element/ action not required

Regarding the amount of entities the protocol involves, the use of IdPs, Hosts and AMs is particularly noticeable in the initialization (Ph1) because relationships between all entities that interact along the protocol are established in this phase.

Besides, a significant amount of entities come into play when accessing data of an indirect contact (Ph4), that is, the longer the relationship being processed, the higher the number of involved entities.

Encryption is applied for a couple of issues. On the one hand, in the initialization (Ph1) resources are encrypted and uploaded to chosen Hosts. On the other hand, encryption protects the delivery of claims. AMs encrypt data involved in requested claims and IdPs encrypt such requested data to be sent to AMs. Besides, it should be noticed that the number of encryptions increases when accessing data of an indirect contact (Ph4) because more claims are requested.

Related to encryption, decryptions are executed at claims management and at resources acquisition. In the end of this Section and associated with data exposure minimization, cryptographic operations are deeply analysed.

Signatures are other elements at stake. They are applied to verify the chain of trust which is created between entities that interchange messages. Signatures are performed by AMs when requesting claims, by IdPs when delivering claims and by WBSNs when sending messages to other WBSNs. Again, the number of signatures increases when accessing data of an indirect contact (Ph4) because more claims and interactions among WBSNs are carried out. Nonetheless, the number of signatures when accessing data of a direct (Ph3) or of an indirect contact (Ph4) in the best case is significantly lower than in the worst case, namely, 2 and $2{\cdot}(N+1)$ in the best case and 16 and $16{\cdot}(N+1)$ in the worst case respectively.

Following expectations, the number of signatures verification is equivalent to the number of signatures. In general, IdPs, AMs and WBSNs create signatures and IdPs and AMs verify them.

Last but not least, the amount of messages involved in eU+F is remarkable. The number of interchanged messages when accessing data of an indirect contact (Ph4) is specially significant. However, it can decrease when tokens and claims are reused because the reuse avoids requesting tokens to AMs and claims to IdPs. In

particular, $50{\cdot}(N+1)$ messages are interchanged in the worst case and $18{\cdot}(N+1)$ in the best case when accessing an indirect contact located at length N.

Analysing the management of SoNeUCON$_{ABC}$ features and usage control eU+F can be extended to manage all WBSN features, namely, *direction*, *flexible elements is access control policies*, *cliques*, *distance*, *common-contacts* and *multi-paths*, and usage control (recall Section 8.5).

On the one hand, the management of all WBSN features involves the definition and evaluation of access control policies according to $SoNeUCON_{ABC}$ policy language (Section 4.3). Then, the construction of rt and the latter verification of access control policies are the points of analysis. However, the evaluation of access control policies can be compared with the study presented in Section 9.1.2.2 and then, the construction of rt in a decentralized architecture like the one proposed in eU+F, is the subject to study.

The TW constructing rt increases according to the sum of all visited users at each path length, that is, $\sum_{i=1}^{K}(\eta^i)$ where η refers to the average number of users' contacts and K corresponds to the path length from 1 to 6 (recall Section 9.1.2.1). The analysis of rt construction is depicted in Table 9.14. From this table, it is noticed that constructing rt requires a significant number of entities, $2{\cdot}\sum_{i=1}^{K}(\eta^i))+1$ in particular. It involves an IdP and a WBSN per users' contact, as well as the AM of the administrator (the owner of the requested resource or identity data) to evaluate access control policies. Then, the number of encryptions corresponds to the number of users' contacts, such that all IdPs encrypt claims and the AM of the administrator verifies them. Likewise, the number of interchanged messages is also remarkable, that is $4 + 6{\cdot}\sum_{i=1}^{K}(\eta^i)$-1. They are interchanged among the AM of the administrator and all appropriate WBSNs and IdPs of users' contacts.

As a result, although eU+F may allow the construction of rt, its application would be impractical in the majority of cases, because the amount of entities and messages that this process involves, would result in an intolerable waiting time for

Table 9.14: eU+F theoretical evaluation: full support of $SoNeUCON_{ABC}$

Phases	Entities	Encryptions	Decryptions	Signatures	Signatures verification	# Messages
rt construction	$2 \cdot \sum_{i=1}^{K}(\eta^i)) + 1$	$\sum_{i=1}^{K}(\eta^i)$	$\sum_{i=1}^{K}(\eta^i)$	$2 \cdot \sum_{i=1}^{K}(\eta^i) - 1 + 1$	$2 \cdot \sum_{i=1}^{K}(\eta^i) - 1 + 1$	$4 + 6 \cdot \sum_{i=1}^{K}(\eta^i) - 1$
Attributes management	I+2·I|I|TTP	I | TTP	I | TTP	I | TTP	I | TTP	I+2·I|I|TTP

I: # of IdPs of a user | K: path length $\leqslant 6$ | TTP: # of TTPs of a user
| : connector OR | η: average number of contacts per user

information retrieval, that is greater than 2,000 ms (Section 9.1.2.3). For instance, regarding *rt id* = 9, which is composed of 27,114 explored nodes and depicted in Table 9.4 (Section 9.1.2.3), the TW of performing the enforcement of P1 is 712 ms (712 ms the construction of *rt* and <1 ms the evaluation of P1 in *rt*). As explored nodes are 27,114, the amount of interchanged messages would be 162,682. Therefore, even assuming that the minimum TW for interchanging a message between a pair of entities is 0.1 ms, the TW of performing the enforcement of P1 would take 16,268 ms and it highly exceeds the tolerable waiting time for information retrieval.

On the other hand, the analysis of attributes updates, additions or deletions, related to usage control, is also depicted in Table 9.14. Contrary to the construction of *rt*, attributes updates can be easily managed in eU+F. In case of $ATT(S)$ or $ATT(E)$ updates, IdPs, TTPs and IdPs and a pair of WBSNs per IdP become involved. Likewise, Hosts notify $ATT(O)$ updates.

Note that policies updates, additions or deletions are not analysed because the process only involves an AM which, after identifying changes, re-evaluate policies accordingly (recall Section 8.5).

Data exposure minimization analysis An analysis of the cryptographic alternatives described in Section 8.4 is performed distinguishing the acquisition of identity data and the acquisition of resources. Results are presented in Table 9.15. Note that this study focuses on cryptographic matters and it is not attached to the rest of eU+F messages.

Regarding entities involved in *Traditional PKC* and *IBE-based PKC*, the latter technique applies a new group of entities called IBE authorities.

Table 9.15: eU+F theoretical evaluation: data exposure minimization

Phases	Entities	Encryptions	Decryptions	Signatures	Signatures verification	# Messages
Traditional PKC	-	2	3	2	2	3
FOAF file acquisition	-	1	1	1	1	-
Resources acquisition	-	1	2	1	1	3
IBE-based PKC	$C_1 \cap C_2$	2	3	-	-	3
FOAF file acquisition	C_1	1	1	-	-	-
Resources acquisition	C_2	1	2	-	-	3

C_x: set x of # of IBE authorities - : an element/ action not required

According to encryption, both techniques require the same number of operations. IdPs create FOAF files and encrypt them once delivered. Likewise, resources acquisition involves the encryption of resources decryption keys.

On the other hand, the number of decryptions acquiring FOAF files and resources differs. FOAF files acquisition simply focuses on decrypting requested files. By contrast, resources decryption requires, first, decrypting the resources decryption key and subsequently, applying this key to decrypt the resources.

In relation to signatures, they are only applied in *Traditional PKC* when acquiring requesters' public keys. These keys are signed and delivered by requesters' IdPs to be properly verified by administrators' IdPs.

Finally, both techniques involve the interchange of three new messages. These messages are used to get resources decryption keys.

9.5.2 Experimental evaluation

The experimental evaluation corresponds to the analysis of eU+F through the extension of the prototype developed to evaluate U+F (Section 9.4.2.1). First, Section 9.5.2.1 briefly describes the developed prototype. Second, Section 9.5.2.2 presents the experimental results regarding the measurement of the protocol temporal workload and its comparison with Facebook and MySpace.

9.5.2.1 eU+F prototype description

The prototype developed to evaluate eU+F is an extension of the one developed to evaluate U+F. Consequently, WBSNs are referred as FriendBook+$_{v1}$ and

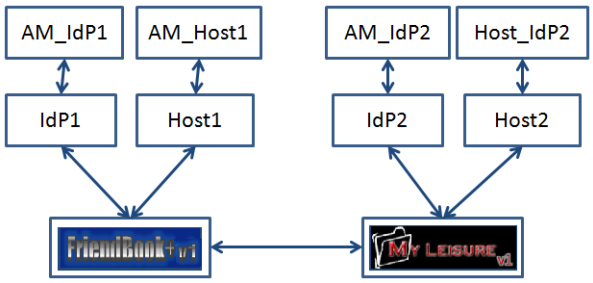


Figure 9.10: Designed eU+F prototype architecture

MyLeisure$_{v1}$, being verified that a user enrolled in FriendBook+$_{v1}$ can access data in MyLeisure$_{v1}$. For simplicity reasons, the prototype only works with direct relationships, and obtained performance results allow to get estimated figures of performance of the protocol for indirect relationships. Indeed, noticed in Section 8.3.2.2, an indirect relationship can be defined as multiple direct ones.

Concerning the prototype architecture, depicted in Figure 9.10, apart from establishing access control policies, Alice and Bob establish resources decryption keys in IdP1 and IdP2 and encrypt resources in Host1 and Host2 respectively.

Analogous to U+F evaluation, applied technologies are J2EE and J2SE 1.6, Glassfish 3.0.1, MySQL 5.2.27 and Firebug 1.7.3. Likewise, photos have a size between 200kb and 300kb. On the other hand, in respect to cryptographic algorithms, the Traditional PKC scheme is implemented (Section 8.4.1). In relation to symmetric cryptography, AES 128 is used to encrypt/ decrypt resources (photos). By contrast, RSA 2048 is the applied asymmetric cryptography algorithm. Besides, it is assumed that each user owns a certificate and a private key of length 2048 bytes.

9.5.2.2 eU+F temporal workload

The temporal workload of the protocol is measured in this Section. More specifically, the access to personal identity data (profile), the access to a personal resource (photo), the access to a direct contact identity data (profile) and the access to a direct contact resource (photo) are analysed. With these results, the workload

Table 9.16: Analysing the reuse of data in eU+F

		Worst case	Best case	% reuse	Avg. Reuse %		$\mho_{maxReuse}$
# Messages	(Ph2)Log-in	24	18	25			
	(Ph3)Access direct contact	50	18	64	51		
	(Ph4)Access indirect contact	50	18	64		59.87	0,41 (1-59%)
Signatures	(Ph2)Log-in	4	2	50			
	(Ph3)Access direct contact	16	2	87.5	68.75		
	(Ph4)Access indirect contact	16	2	87.5			

of accessing an indirect contact identity data and an indirect contact resource is estimated and analysed.

The total workload of performing any kind of access is measured as the cost of interchanging protocol messages until reached the requested data ($C_{dataAccess}$) multiplied by a parameter $\mho$ (recall it refers to not reused information) plus the cost of performing the required decryptions ($C_{dataDecryption}$), Equation 9.2. Moreover, as in U+F, an analysis regarding possible values of $\mho$ is performed by comparing the number of signatures carried out and the number of messages interchanged in the worst case (no elements are reused) and in the best case (all possible elements are reused) when a user logs in to a WBSN (Ph2), a user accesses a direct contact's data (Ph3) and a user accesses an indirect contact's data (Ph4) (recall Table 9.13). Considering that reusing is unachievable regarding the acquisition of the personal identity data because it is the first requested data, the performed analysis, presented in Table 9.16, shows that 68.75% of signatures and 51% of messages are reused, concluding that, on average, the maximum level of reuse is 59.87%. Consequently, three values of $\mho$ are considered, $\mho$ is 1 when not a single piece of data is reused, 0.70 when 50% of data is reused and 0.41 when all data is reused, that is 59.87%. The workload has been measured as the average of 10 executions and executions have been carried out without supposing the reuse of any element ($\mho$=1).

$$C_{total} = C_{dataAccess} \cdot \mho + C_{dataDecryption} \tag{9.2}$$

According to these features, plot presented in Figure 9.11 depicts the total workload of accessing to the profile and to a photo of a user registered in WBSN1

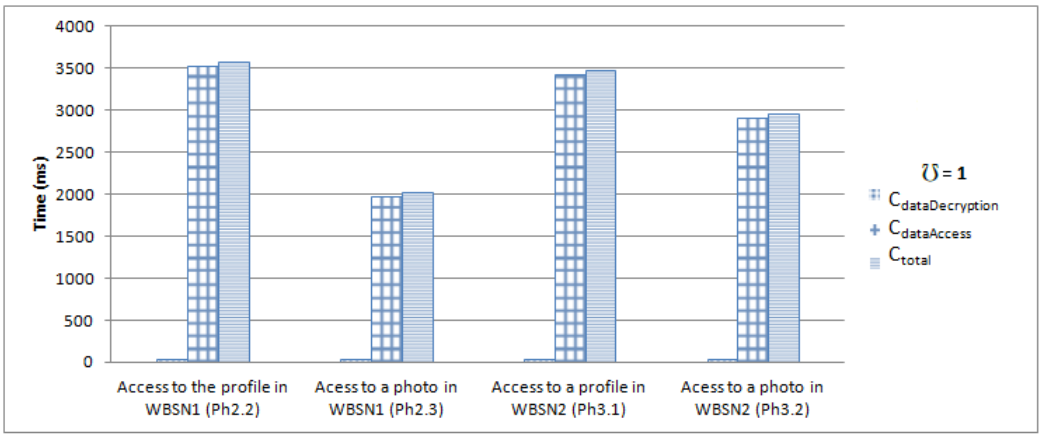


Figure 9.11: Temporal costs comparison

(FriendBook+$_{v1}$) and to the profile and to a photo of a direct contact enrolled in WBSN2 (MyLeisure$_{v1}$). Furthermore, the workload regarding the individual costs of $C_{dataAccess}$ and $C_{dataDecryption}$ in the worst case, that is, ʊ=1, is also presented. It is identified that $C_{dataAccess}$ implies a high workload while, $C_{dataDecryption}$ is rather small. Although cryptographic operations depend on the applied algorithm, the decryption scheme draws satisfactory results. Decryptions take 86.83 ms on average, 83.83 ms for profiles and 89.83 ms for photos. Recalling that profiles are encrypted with an asymmetric algorithm and photos with a symmetric one, and considering the fact that asymmetric algorithms are generally slower than symmetric algorithms, results may be cause because photos are bigger in size than profiles and then, the workload is rather similar.

Analysing the same features as in the previous plot, except for the access to the profile of a user registered in WBSN1 (because reuse is not possible), Figure 9.12 presents workloads in regard to different ʊ values. It is remarkable that to achieve successful results, reuse is a matter of concern. Besides, as expected, interoperability between WBSN1 and WBSN2 increases the workload. The difference between accessing to a particular data in WBSN1 and accessing to WBSN2 is 1,191.90 ms when ʊ=1, 834.33 ms when ʊ=0.70 and 488.68 ms when ʊ=0.41.

On the other hand, the establishment of indirect relationships is a challenging

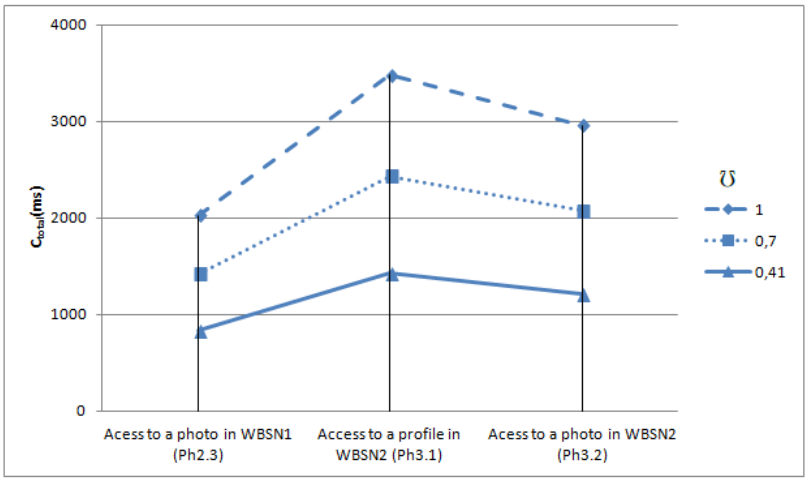
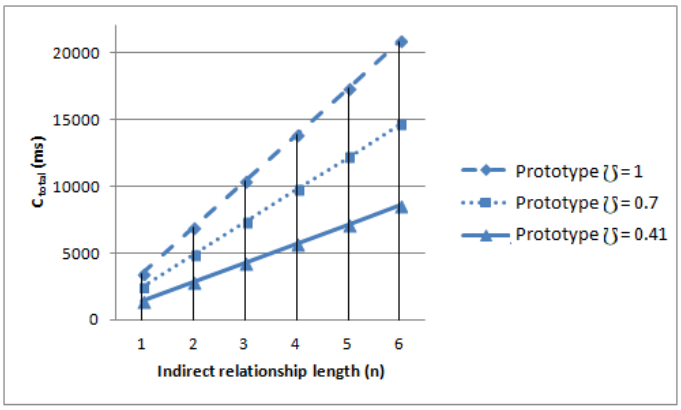


Figure 9.12: General temporal workload

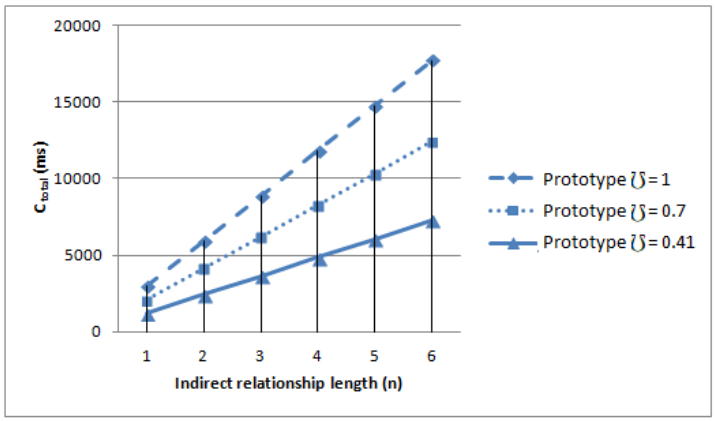
goal achieved in eU+F but not implemented in the prototype. Nonetheless, assuming that an indirect relationship is composed of direct ones, the workload is estimated as the cost of data acquisition multiplied by ʊ and by the length of indirect relationships (n) plus the cost of data decryption (see Equation 9.3). Therefore, plots presented in Figures 9.13(a) and 9.13(b) show the estimated workload in respect to different values of ʊ and n, given that n is bounded to 6 due to theoretical studies pointed out in Section 7.3.1.3.

$$C_{total} = C_{dataAccess} \cdot ʊ \cdot n + C_{dataDecryption} \tag{9.3}$$

From the analysis it is identified that according to the longest indirect relationship (n=6) in the worst case, ʊ=1, about 20,851 ms are needed to access a chosen profile and about 17,744 ms to a chosen photo. On the contrary, in the best case, ʊ=0.41 for n=6, about 8,549 ms and 7,275 ms are taken to access a profile and to a photo respectively. Nevertheless, the reuse of data, for instance a user's credential, is highly probable and the average workload can be taken as a representative measure. In particular, for ʊ=0.7, workload is 2,432 ms for n=1 and 14,596 ms for n=6 to access a profile and 4,140 ms for n=1 and 12,421 ms for n=6 to access a photo.



(a) Access a profile



(b) Access a photo

Figure 9.13: Estimation of temporal workload for indirect relationships

A final remark is that parallelization is not an issue to consider because the operation flow needs to be sequential, i.e. to get a resource you firstly need a token. By contrast, precomputation, referred to as the process of getting tokens or claims in advance, could be feasible. Nonetheless, obtaining claims may involve a significant amount of work and then, the request of tokens and claims on demand is expected to be a better alternative. Indeed, as pointed out in this evaluation, the reuse of tokens and claims is the best choice to improve the TW.

9.5.2.3 Temporal workload comparison

The prototype (FriendBook+$_{v1}$ and MyLeisure$_{v1}$) workload is compared against Facebook and MySpace. The same set of photos uploaded to this pair of WBSNs

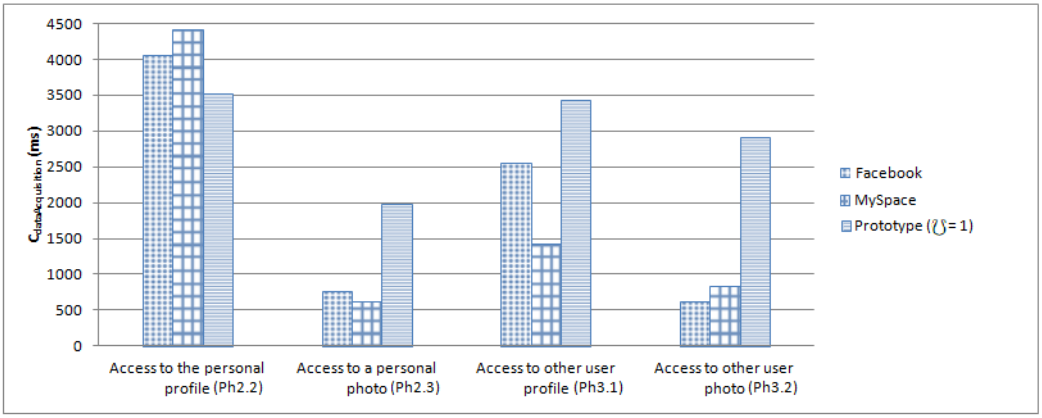
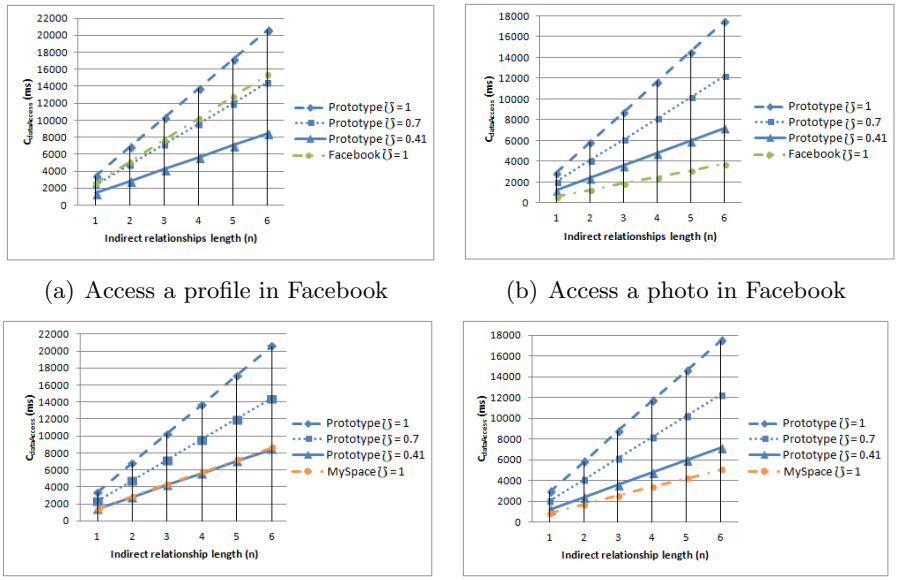


Figure 9.14: Prototype, Facebook and MySpace total cost comparison

is uploaded to FriendBook+$_{v1}$ and MyLeisure$_{v1}$. Then, the workload of accessing the personal profile and a personal photo of a user enrolled in FriendBook+$_{v1}$ or in MyLeisure$_{v1}$ and the workload of accessing the profile and a photo of a user enrolled in the opposite WBSN, are measured.

Based on the workload of Facebook and MySpace, applied in U+F evaluation (Section 9.4.2.1), and the measured workload of FriendBook+$_{v1}$ and MyLeisure$_{v1}$, the comparison is presented in Figure 9.14. According to Equation 9.2, as in current WBSNs cryptographic techniques are not applied, the analysed workload is bounded to $C_{dataAccess}$.

According to the protocol phases, this study compares temporal costs for accessing to the profile of a user when he logins in a WBSN (Ph2.2), to one of his photos (Ph2.3) and to the profile and to a photo of a direct contact (Ph3.1 and Ph3.2 respectively). These costs are respectively 3,529 ms, 1,878 ms, 3,432 ms and 2,913 ms for the prototype, whereas they are 4,423 ms, 626 ms, 1,438 ms and 842 ms in the case of MySpace and 4,052 ms, 766 ms, 2,556 ms and 624 ms in the case of Fakebook. It is noticed that, as in U+F, accessing to a MySpace personal profile produces the highest workload. However, either Facebook or MySpace workload accessing to personal photos and to the profile and photos of a direct contact is lower than in the proposed prototype. This issue is not surprising because reach-



(a) Access a profile in Facebook

(b) Access a photo in Facebook

(c) Access a profile in MySpace

(d) Access a photo in MySpace

Figure 9.15: Facebook and MySpace indirect relationships comparison

ing interoperability is a challenging issue which expectedly produces a workload increase. Moreover, already mentioned, the implementation is a prototype which is far from being developed by powerful software and deployed on optimized hardware mechanisms.

On the other hand, the relevance of indirect relationship management requires its analysis. First of all, to establish comparable parameters and even not being currently possible, it is assumed that Facebook and MySpace allow the establishment of indirect relationships of a maximum length of six. Then, given Equation 9.3, the workload is calculated considering $C_{dataAccess}$ multiplied by each relationship length. Besides, note that Facebook and MySpace, as far as we know, do not reuse data because elements like tokens or claims are not used. Concerning this feature, Figure 9.15 presents achieved results. On the whole, it is remarkable that Facebook indirect relationship workload is similar to the developed prototype accessing to a profile when 50% of elements are reused (υ=0.7) and lower than the prototype when accessing to a photo. By contrast, results show that, when accessing to a profile MySpace workload is close to $\upsilon = 0.41$ in the prototype. However, the TW

accessing to a photo in MySpace is also lower than in the prototype although all possible elements are reused, that is $\mho = 0.41$.

The final point to highlight is that, also analogous to U+F, the proposed prototype has been executed on a local network, thereby being possible a workload increase in a real scenario. However, their execution and implementation are not supported by powerful techniques and mechanisms.

9.6 Evaluation of U+F vs eU+F

Apart form evaluating U+F and eU+F independently, the performance of a comparative assessment is worth studying. Both protocols have been evaluated under the same bases and their comparison can be easily performed, either theoretically or empirically as Sections 9.6.1 and 9.6.2 respectively present.

9.6.1 Performance analysis comparison

UMA and the FOAF project lay the bases of U+F and eU+F. In consequence, both protocols share a significant set of elements. Table 9.17 compares the number of involved entities, signatures, signatures verifications and messages interchanged in each protocol. Identified from the table, in eU+F more signatures are required, a new group of entities is added (AM Certification Authorities, AM_CAs), C_A, and a new execution phase for indirect relationships management is developed.

Nonetheless, despite the amount of similarities between eU+F and U+F, messages content differs to a great extent. Comparing interchanged messages in eU+F with those interchanged in U+F, the following features are distinguished:

- Tokens and claims requests are signed by the appropriate AMs.
- Requested claims are encrypted by AMs and decripted by IdPs. Conversely, IdPs encrypt requested claims and AMs decrypt them.
- Messages signed by AMs include the AM certificate serial number.

Table 9.17: Theoretical comparison U+F vs eU+F

Phases		Entities	Signatures	Signatures verification	# Messages
		Login			
FOAF file acquisition	U+F	$3+C_I + C_{WBSN}$	1	1	12
	eU+F	$3+C_I + C_{WBSN} + C_A$	2	2	12
Resource acquisition	U+F	$3+C_I + C_{WBSN}$	1	1	12
	eU+F	$3+C_I + C_{WBSN} + C_A$	2	2	12
		Access to direct contact			
FOAF file acquisition	U+F	$6+C_I + C_{WBSN}$	6	6	25
	eU+F	$6+C_I + C_{WBSN} + C_A$	8	8	25
Resource acquisition	U+F	$7+C_I + C_{WBSN}$	6	6	25
	eU+F	$7+C_I + C_{WBSN} + C_A$	8	8	25
		Access to indirect contact			
FOAF file acquisition	U+F	-	-	-	-
	eU+F	$3{\cdot}(N+1) + C_I + C_{WBSN} + C_A$	$8{\cdot}(N+1)$	$8{\cdot}(N+1)$	$25{\cdot}(N+1)$
Resource acquisition	U+F	-	-	-	-
	eU+F	$3+4{\cdot}N + C_I + C_{WBSN} + C_A$	$8{\cdot}(N+1)$	$8{\cdot}(N+1)$	$25{\cdot}(N+1)$

N: (# of users in the relationship)-1, N>1
C_x: # of IdP_CAs, AM_CAs and WBSN_CAs, where x is I, WBSN or A regarding the type of CA
- : an element/ action not required

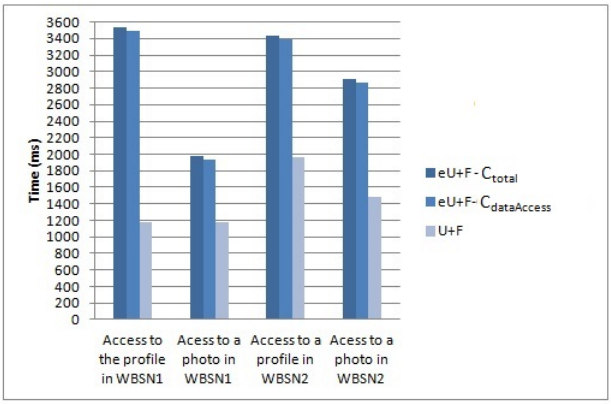
- Depending on the applied cryptographic approach (recall Section 8.4), the interchanged of the decryption key in the Traditional PKC scheme or the decryption key creation in the IBE-based PKC scheme, is required.

- Resources and identity data are delivered encrypted to be decrypted at users' browsers.

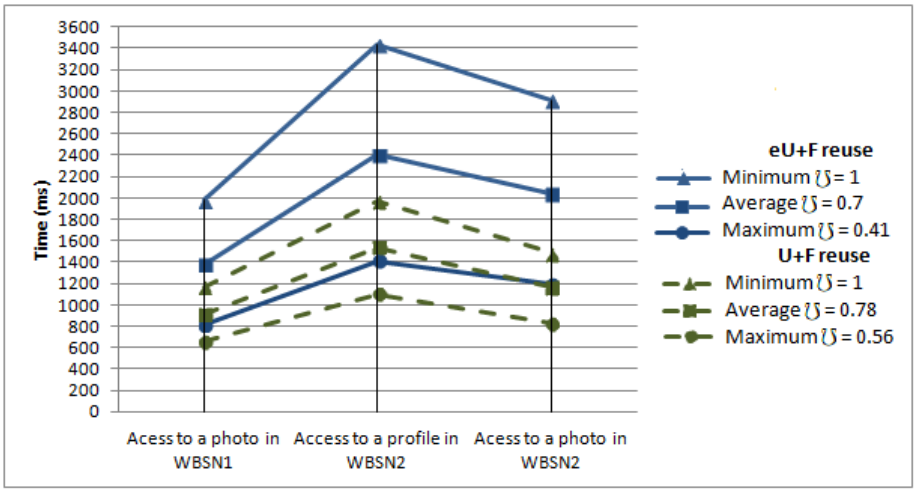
9.6.2 Temporal workload comparison

Recalling the empirical evaluation of U+F and eU+F, this Section compares their workload and reused elements.

On the one hand, Figure 9.16(a) depicts temporal workload of U+F and eU+F. As expected, temporal workload of eU+F is higher than that of U+F, even determining that identity data and resources decryption slightly impacts on the general workload (44 ms more on average). Comparing eU+F $C_{dataAcquisition}$ with U+F, accessing to a profile and a photo in a WBSN, WBSN1, and accessing to a profile and



(a) Execution phases



(b) Elements reuse

Figure 9.16: Comparing U+F vs eU+F

a photo in other WBSN, that is, WBSN2, differ between 2,311 ms, 759 ms, 1,420 ms and 1,378 ms respectively. According to previous theoretical study, workload results are produced by eU+F claims management and the applied cryptographic scheme. On the one hand, claims are encrypted by AMs and IdPs. On the other hand, the implemented Tradictional PKC scheme (recall Section 8.4.1) involves the signature and the interchange of the requester's DK key and the encryption of requested identity data and of resources decryption keys.

On the other hand, the reuse of elements is also compared. Interestingly, the maximum percentage of reuse in eU+F (59%, ℧=0.41) is higher than that of U+F

(44%, $\mho$=0.56), being this issue related to indirect relationships management. Requesting data in indirect relationships involves the interchanged of many messages, being multiple of them reused and thus, the workload reduced. For this reason, eU+F workload is the highest one when $\mho$=1. However, the reuse of all possible elements in eU+F ($\mho$=0.41) can be compared with the average rate of reuse in U+F ($\mho$=0.78), differing 104 ms, 128 ms and 33 ms accessing to a photo in a WBSN, WBSN1, and to a profile and a photo in other WBSN, WBSN2, respectively. Similarly, when the reuse is maximum in both protocols ($\mho$=0.41 in eU+F and $\mho$=0.56 in U+F), temporal workload in eU+F differs from U+F on 154 ms, 304 ms and 360 ms in each case.

9.6.3 Protocols adequacy analysis

The performance and workload studies previously presented, point out the simplicity and speed of U+F and the security features of eU+F. Bearing in mind that Hosts and WBSNs are considered trusted entities in U+F but not in eU+F, the appropriateness of these protocols depends on each particular context.

Concerning the use of U+F, its use is specially adequate when managed data is not confidential and direct relationships are enough to meet expectations. For instance, in WBSNs like LinkedIn, a professional WBSN where users present their working experience and may use it for seeking employment, the preservation of users privacy is not the main requirement and the establishment of direct relationships is desirable in most of cases. Then, in these situations, when data wants to be also applied in other WBSN, i.e. Facebook, U+F would be the best choice.

On the other hand, personal data should be protected by the use of eU+F. WBSNs like Facebook or Badoo, in which, for instance, many personal photos or even personal embarrassing photos are uploaded, should apply eU+F. This protocol provides interoperability, indirect relationships management and protection against honest-but-curious servers, thereby developed on the bases of users privacy protec-

tion. Therefore, the increase of temporal workload produced by eU+F should be considered balanced by provided features.

Chapter 10

Conclusions

This Chapter contains the thesis conclusions and final remarks, and summarizes the achieved contributions (Section 10.1). A critical discussion on the developed work is also presented (Section 10.2). Additionally, future research directions that derive from the thesis results are proposed (Section 10.3).

10.1 Conclusions and summary of contributions

This thesis focuses on providing fine-grained access control management between different WBSNs in a privacy preserving way. Improving access control facilitates and protects users data management. It reduces threats caused mainly by three issues. First, limited access control procedures prevent users from expressing all their preferences and controlling co-owned data. Second, the burden of managing access control in all WBSNs in which users are enrolled may become tedious and cumbersome. Lastly, WBSN provides stored data and they can use them for unnoticed or unauthorized purposes.

There are a lot of previous approaches concerning access control management in WBSNs, as well as some mechanisms candidates for achieving interoperability among different WBSNs. However, existing proposals neither are focused on expressive access control management, nor on interoperability, reusability and prevention of unauthorized data exposures. To address these issues, this thesis proposes an expressive usage-based Access Control Model (ACM), together with its administrative functions, that lay the bases of a co-ownership mechanism and a pair of

protocols to attain interoperability and reusability between WBSNs.

The first contribution of this thesis consists of the definition of the mentioned ACM (Chapter 4) and the attached administrative model (Chapter 5), called $SoNeUCON_{ABC}$ and $SoNeUCON_{ADM}$ respectively. $SoNeUCON_{ABC}$ is an expressive usage control model for the social networking field. It allows managing a total of six WBSN features (*distance*, *common-contacts*, *clique*, *multi-path*, *direction* and *flexible attributes*) along the whole usage process. It has been theoretically identified that all features are addressed and empirically, that the model's implementation is feasible and applicable to the majority of cases. A proof of concept system has been developed in this regard. Considering 2,000 ms the tolerable waiting time of WBSN users for information retrieval, results show that the enforcement of policies without cliques is satisfactory if explored nodes do not exceed about 200,000 having 200 relationships per node. Besides, the enforcement of policies with cliques remains successful if less than about 30,000 nodes and 200 relationships per node are explored. As a result, the appropriateness of the model for the WBSN context is highlighted.

In addition, $SoNeUCON_{ADM}$, the administrative model for $SoNeUCON_{ABC}$, supports administrative tasks concerning the identification of who is involved in administrative issues and how they are performed. $SoNeUCON_{ADM}$ has been assessed against a pair of administrative access control models (RBAC and $UCON_{ABC}$) to ensure that it successfully addresses all identified administrative tasks.

Looking for co-ownership management and based on $SoNeUCON_{ABC}$ and $SoNeUCON_{ADM}$, the second contribution of this thesis is a mechanism called Co-owned Personal Data, CooPeD (Chapter 6). It is a novel mechanism based on managing decomposable objects according to owners and co-owners privacy preferences, such that all these preferences are completely satisfied and thus, all users' privacy is preserved. Using the same proof of concept as the one applied in $SoNeUCON_{ABC}$,

the access control enforcement process in CooPeD is analysed. It is concluded that if a pair of users is related to an structure rt with direct relationships, a maximum of 34 policies can be evaluated per object without exceeding the tolerable waiting time of WBSN users for information retrieval. By contrast, if an rt with indirect relationships is at stake, a maximum of 4 policies can be evaluated per object. Furthermore, the developed survey study, fulfilled by 206 people worldwide, shows that co-ownership is worth studying and that 76.2% of respondents are potential users of CooPeD.

Apart from previous issues, the amount of WBSNs and the management of data in all of them may lead to privacy problems. Users have to manage data in all WBSNs in which they are enrolled. Thus, based on a simplified version of $SoNeUCON_{ABC}$, a couple of protocols to manage interoperability and reusability in WBSNs comprise the last contribution. One of the protocols, called UMA+FOAF social network protocol (U+F), provides interoperability and reusability of identity data, resources and access control policies between different WBSNs among users directly connected (Chapter 7). Extending this protocol, Extended UMA+FOAF social network protocol (eU+F) is proposed to protect WBSN data against service providers and to facilitate indirect relationship management (Chapter 8). Both protocol have been theoretically and empirically evaluated. The satisfaction of all established requirements has been theoretically studied, as well as the effect of reusing tokens and claims in every protocol's phase. Besides, a prototype, composed of the WBSNs FriendBook+ and MyLeisure, shows the feasibility of implementing U+F and eU+F, as well as it helps to measure the Temporal Workload (TW) of each protocol's phase in comparison with Facebook and MySpace. Results show that, accessing to a contact's photo or a contact's profile in Facebook or MySpace produces lower TW that accessing a photo or a profile of MyLeisure from FriendBook+. Nonetheless, the TW accessing a photo or a profile in Facebook and MySpace is comparable with the prototype when some elements (tokens and claims

in particular) are reused. Indeed, interoperability and decentralization, that is, the fact of locating resources in host, identity data in IdPs, etc., produce benefits in exchange for TW.

Proposed contributions have been published or submitted to several conferences and journals. Table 10.1 presents the relationship between contributions and publications. Note that references of Table 10.1 refer to the list of publications shown in Appendix B

Table 10.1: Administrative tasks comparison

Contributions	**Publications**
C1: An expressive usage control model for WBSNs and its administrative model.	[2], [3], [5], [7]
C2: A mechanism to manage co-ownership of decomposable objects in WBSNs.	[8]
C3: A mechanism to reach interoperability and reusability among different WBSNs that also minimizes unauthorized data exposures.	[1], [4], [6]

10.2 Critical analysis on the developed work

$SoNeUCON_{ABC}$ usage control model lays the foundations of all contributions of this thesis and it focuses on providing expressive access control management in the social networking field. Nonetheless, the adequacy of the proposed policy language in regard to WBSN users' expectations and likes should be assessed.

A question arises over using $UCON_{ABC}$ to create $SoNeUCON_{ABC}$ instead of other model directly developed for the WBSN field (see Section 2.2). At first sight, the relevance of relationships management supports the need of RelBAC models. By contrast, although relationships are key elements of WBSNs, studies reveal that other elements like users attributes or object attributes are essential too (Section 2.5). Therefore, paying attention to WBSN demands, attributes management becomes essential and $UCON_{ABC}$ is a challenging model in this regard. Furthermore, the application of an usage control model like this is an interesting issue because ac-

cording to current trends, the management of access control along the whole usage process is a desirable requirement [20].

With regard to the proposed administrative model, $SoNeUCON_{ADM}$, its implementation either in a real or in a simulated environment is expected in future work. Specifically, the presented theoretical evaluation does not study users satisfaction.

CooPeD, the proposed co-ownership management mechanism, works over decomposable objects whose management involves a pair of limitations. On the one hand, being decomposable requires the recognition of objects parts. Thus, restrictions of recognition tools should be overcome. Owners can manually decompose objects but automatic tools are preferable. For example, users who are dancing and embracing to each other are hard to be appropriately decomposed. On the other hand, the management of objects parts that belong to multiple users, even being out of the scope of this proposal and a matter of future work, has to be considered. For instance, assuming an image of a married couple in front of their car, the image is decomposed in three parts, namely, two parts in regard to the couple and other in regard to the car. In this scenario who and how access control preferences over the car part are managed should be distinguished.

Concerning the achievement of interoperability and reusability among different WBSNs, U+F is developed and extended in eU+F to reach data exposure minimization and indirect relationships management. Both protocols are based on a simplified version of $SoNeUCON_{ABC}$ which avoids the construction of rt. However, how eU+F can be extended to fully support $SoNeUCON_{ABC}$ has been discussed, namely, managing usage control and all features $SoNeUCON_{ABC}$ deals with (recall Section 8.5). In particular, related to usage control, when attributes or policies updates, additions or deletions are detected, the access control enforcement process is repeated accordingly. Nonetheless, repeating this process may affect the protocol performance, thereby being indispensable a more rigorous analysis in terms of usage control management. On the other hand, eU+F may allow the management of

all features $SoNeUCON_{ABC}$ deals with when rt is constructed and policies evaluated over it. However, the construction of rt is quite complex and a careful study regarding the best way to enforce its construct is needed.

In addition, a pair of prototypes (one based on the other) have been developed for the evaluation of U+F and eU+F. They allow the access to data of a WBSN from another, keeping identity data, resources and access control policies decentralized. The prototypes try to reproduce real WBSNs but they are not deployed in a real environment which may affect the results' accuracy.

In what concerns trust management, U+F and eU+F assume that it is performed in a successful and reliable manner. Certification Authorities (CAs) are assumed to issue certificates to trusted entities (either IdPs, WBSNs or AMs) without being established the meaning of "trusted entity".

Regarding trust models, the one proposed in U+F is rather restrictive and it is successfully improved in eU+F. First of all, eU+F assumes trusted IdPs. Entities in charge of storing personal data are presumably trusted and users privacy may be violated otherwise. Specifically, IdPs store identity data and resource decryption keys and then, if they act maliciously data of all WBSN users may become compromised. Similarly, other assumption is that AMs are also trusted. This assumption also helps to protect users' privacy. Information received by AMs is used to evaluate access control policies and get tokens. However, tokens have to be signed by the appropriate WBSN to be presented to the right Host or IdP and to lastly achieve requested data. In the opposite situation, that is, in case untrusted AMs come into play, they may deliver tokens to WBSNs and these entities would be in the position of, again, acquiring data illegitimately.

Also related to eU+F, impersonations may exist because OAuth, which is the underlying protocol, does not prevent them (recall Section 8.3.1.2). To manage this issue this protocol delivers encrypted data and then, impersonations cannot affect users privacy. However, this kind of attack may impact on the protocol, e.g.

performance, and much more work should be performed in this regard.

As a final point of analysis, the management of data after their delivery is worth mentioning. This matter, referred to as *sticky policies* in regard to Gates *et al.* requirements, is addressed in this thesis by constructing $SoNeUCON_{ABC}$ on the bases of $UCON_{ABC}$ usage control model. Nonetheless, although this model involves usage control, it still remains much to be done until the practical realization of this requirement.

10.3 Challenges and future research lines

Proposals presented in this thesis are opened to new research developments which may contribute to complement this work, as well as to provide a wide view about access control in WBSNs.

Regarding $SoNeUCON_{ABC}$, implementing a more efficient cliques evaluation algorithm would be desirable, as well as a large-scale scalability analysis. Furthermore, the identification of a holistic extensible and unified catalogue of $ATT(S)$, $ATT(O)$ and $ATT(E)$ used in current WBSNs would be attractive.

Besides, the analysis of the complexity and the amount of user actions involved in access control policies construction is other future research line. Related to this issue, proposed features have been identified from literature and, though researchers seem to be far from reality, studies support the appeal of fine-grained access control systems [178, 179]. Nevertheless, it still remains as an open research issue the usability of the policy construction based on these features. Indeed, an interesting approach would be the development of automatic tools to establish access control policies on the bases of $SoNeUCON_{ABC}$, thus easing the complexity of policies management. The study of recommender systems like [180] would be the first step. This recommender makes suggestions according to attributes of the object to recommend, attributes of the requester and the "ratings" over the object that direct contacts have performed.

One last point related to $SoNeUCON_{ABC}$ is that despite being a privacy-preserving model, the protection of user relationships requires much more work [181]. Although users attributes are unknown, the network structure in terms of attributes can be currently inferred.

In what concerns $SoNeUCON_{ADM}$, the main future step is the management of temporal delegations. Additionally, related to $SoNeUCON_{ABC}$ and $SoNeUCON_{ADM}$, both models could be extended to allow the modification of $ATT(O)$ of uploaded objects either by owners or by co-owners.

The study of users curiosity and suspiciousness in CooPeD is other research line. Particularly, unless using sophisticated techniques to hide objects parts, questions such as who/what is under the hidden part on the object? or how can I get to know him/her/it? may arise. Besides, the management of objects like documents, music, videos with audio, etc. is significantly complex and the identification of co-owners should be carefully analysed, starting by an in-depth study of decomposition techniques.

Another future issue is the enhancement of the implemented prototype. Sophisticated techniques to accurately detect users silhouettes should be developed, thereby hiding users with higher precision. Besides, the prototype can be extended to detect not only users but also animals, vehicles, etc., as well as to manage of usage control. Also, the search of concrete scenarios where CooPeD may contribute significantly, e.g. to protect children privacy, is a challenging issue.

Concerning U+F and eU+F protocols, given that the latter is based on the former, open challenges are associated with eU+F. It can be extended in several ways. First and foremost, the developed prototype has to be improved to be deployed in a real environment, including the management of indirect relationships and access control along the whole usage process. Additionally, eU+F has to work over $SoNeUCON_{ADM}$, managing all administrative tasks this model involves.

Continuing with eU+F future challenges, cryptography is a matter of study.

Data exposure minimization is managed through the application of cryptography and, as a preliminary step, the analysis of cryptographic algorithms efficiency is indispensable. To do so, a comparative study of multiple algorithms needs to be performed.

Moreover, the specification of constraints and rules to specify what it is considered a trusted IdP, WBSN and AM is an open issue. The idea is similar to the one proposed by J. Kang *et al.* in [182]. They present the creation of *guardians*, people with a new profession, to protect personal data and it includes a detailed description of legal relations between *guardians* and clients. Other relevant matter is that, currently, protocols abort if a particular AM, IdP or WBSN is not considered trusted (Sections 7.3.2 and 8.3.2). Thus, a dynamic specification of trusted entities is desirable, e.g. requesting the appropriate administrator about the consideration of a new entity as a trusted one.

Besides, although in U+F and in eU+F impersonations attacks are avoided, the modification of tokens to prevent impersonations should be considered, not due to their intended purposes but due to their side effects such as denial of service.

Related to the whole set of proposals, a further step is to work towards reaching a complete protection of users privacy by preventing WBSNs from inferring user relationships. Currently, social relationships can be inferred and though not directly affecting users privacy because subject attributes are protected, countermeasures against this issue are demanding necessities. The work proposed by Carminati *et al.* in [148] which uses certificates to protect relationships, can be taken as a starting point.

Furthermore, as J. Park *et al.* identifies, the distinction between users and sessions is an appealing matter associated with all contributions of this thesis [183]. Policies may be defined in regard to users sessions. For instance, a user opens different sessions from different computers, which means from different IP addresses, and access control mechanisms should provide him with different permissions per

session. Thus, future work runs towards the study of novel approaches to include this issue in the proposed models and mechanisms.

Last but not least, the specification of techniques and mechanisms to control data after their delivery is a tough challenge to deal with. This matter can be somehow addressed by the use of an usage control model. Nevertheless, even some proposals work in this direction [101, 72] and this thesis provides some guidelines, more details are required. Related to this issue, the change of users behaviour, from trusted to untrusted, is an interesting topic. A significant starting point focuses on studying approaches like [43], which presents a technique to monitor users once logged in WBSNs. According to the contributions of this thesis, a behavioural change can be compared with an attribute update and the necessity of revoking granted rights.

Part V

Bibliography and appendices

Bibliography

[1] Machulak, M. P. and Maler, E. L. and Catalano, D. and van Moorsel, A. User-managed access to web resources. In: Proceedings of the 6th ACM workshop on Digital identity management. DIM '10p.35–44.

[2] Harper, R. and Rodden, T. and Rogers, I. and Sellen, A. Being Human: Human-Computer Interaction in the Year 2020. Microsoft Corporation; 2008.

[3] Ellison, N. B. and others. Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication. 2007;13(1):210–230.

[4] Kumar, R. and Novak, J. and Tomkins, A. Structure and evolution of online social networks. In: Link Mining: Models, Algorithms, and Applicationsp.337–357.

[5] Parent, W. A. Privacy, Morality, and the Law. Philosophy and Public Affairs. 1983;12(4):269–288.

[6] Acquisti, A. and Gross, R. Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. In: Privacy Enhancing Technologies. Lecture Notes in Computer Sciencep.36–58.

[7] Hoadley, C. M. and Xu, H. and Lee, J. J. and Rosson, M. B. Privacy as information access and illusory control: The case of the Facebook News Feed privacy outcry. Electronic Commerce Research and Applications. 2010;9(1):50 – 60.

[8] Becker, J. and Chen, H. Measuring Privacy Risk in Online Social Networks. In: Proceedings of W2SP 2009: Web 2.0 Security and Privacy; 2009. .

[9] Oracle-Team. Online security, A Human Perspective. 2011;.

[10] Dey, R. and Jelveh, Z. and Ross, K. W. Facebook Users Have Become Much More Private: A Large-Scale Study. In: Proceedings of SESOC 2012; 2012. .

[11] European Parliament. Directive 95/46/EC of the European Parliament and of the Council; 1995.

[12] United Nations General Assembly. Resolution: The right to privacy in the digital age; 2013.

[13] Dwyer, C. and Hiltz, S. R. and Passerini, K. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In: Proceedings of AMCIS; 2007. .

[14] Bonneau, J. and Preibusch, S. The privacy jungle: On the market for data protection in social networks. In: In The Eighth Workshop on the Economics of Information Security (WEIS 2009); 2009. .

[15] Liu, Y. and Gummadi, K. P. and Krishnamurthy, B. and Mislove, A. Analyzing facebook privacy settings: user expectations vs. reality. In: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. IMC '11. ACMp.61–70.

[16] Hu, V. C. and Ferraiolo, D. and Kuhn, D. R. Assessment of access control systems. US Department of Commerce, National Institute of Standards and Technology; 2006.

[17] Carminati, B. and Ferrari, E. Access control and privacy in web-based social networks. In: International Journal of Web Information Systems. 4p.395–415.

[18] Foresti, S. Preserving privacy in data outsourcing. Springer; 2010.

[19] Fong, P. W. L. and Siahaan, I. Relationship-based access control policies and their policy languages. In: Proceedings of the 16th ACM symposium on Access control models and technologies. SACMAT '11. ACMp.51–60.

[20] Lazouski, A. and Martinelli, F. and Mori, P. Usage control in computer security: A survey. Computer Science Review. 2010;4(2):81 – 99.

[21] Anderson, J. and Stajano, F. Not That Kind of Friend: Misleading Divergences Between Online Social Networks and Real-World Social Protocols. In: Proceedings of the Seventeenth International Workshop on Security Protocols (SPW' 09). Citeseerp.1–6.

[22] PrimeLife-members. Requirements and concepts for privacy-enhancing access control in social networks and collaborative workspaces. 2009;.

[23] Lazer, D. and Pentland, Alex S. and Adamic, L. and Aral, S. and Barabasi, A. L. and others. Life in the network: the coming age of computational social science. Science (New York, NY). 2009;323(5915):721.

[24] Li, J. and Tang, Y. and Mao, C. and Lai, H. and Zhu, J. Role Based Access Control for social network sites. In: Pervasive Computing (JCPC), 2009 Joint Conferences onp.389 –394.

[25] Ding, J and Mo, L. Enforcement of Role Based Access Control in Social Network Environments. In: Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference onp.92 –101.

[26] Giunchiglia, F. and Zhang, R. and Crispo, B. RelBAC: Relation Based Access Control. In: Semantics, Knowledge and Grid, 2008. SKG '08. Fourth International Conference onp.3 –11.

[27] Fong, Philip W. L. Relationship-based access control: protection model and policy language. In: Proceedings of the first ACM conference on Data and application security and privacy. CODASPY '11. ACMp.191–202.

[28] Carminati, B. and Ferrari, E. and Perego, A. Rule-Based access control for social networks. In: Proceedings of the 2006 international conference on On the

Move to Meaningful Internet Systems: AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET - Volume Part II. OTM'06. Springer-Verlagp.1734–1744.

[29] Masoumzadeh, A. and Joshi, J. OSNAC: An Ontology-based Access Control Model for Social Networking Systems. In: Social Computing (SocialCom), 2010 IEEE Second International Conference onp.751 –759.

[30] Park, J. and Sandhu, R. and Cheng, Y. ACON: Activity-Centric Access Control for Social Computing. In: Availability, Reliability and Security (ARES), 2011 Sixth International Conference onp.242 –247.

[31] Kumari, P. and Pretschner, A. and Peschla, J. and Kuhn, J-M. Distributed data usage control for web applications: a social network implementation. In: Proceedings of the first ACM conference on Data and application security and privacy. ACMp.85–96.

[32] Hu, H. and Ahn, G-J. and Jorgensen, J. Multiparty Access Control for Online Social Networks: Model and Mechanisms. IEEE Transactions on Knowledge and Data Engineering. 2012;99.

[33] Squicciarini, A. C. and Xu, H. and Zhang, X. L. CoPE: Enabling collaborative privacy management in online social networks. Journal of the American Society for Information Science and Technology. 2011;62(3):521–534.

[34] Thomas, K. and Grier, C. and Nicol, D. M. unfriendly: multi-party privacy risks in social networks. In: Proceedings of the 10th international conference on Privacy enhancing technologies. PETS'10p.236–252.

[35] Aiello, L. M. and Ruffo, G. Secure and flexible framework for decentralized social network services. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on. IEEEp.594–599.

[36] Shakimov, A. and Lim, H. and Cáceres, R. and Cox, L. P. and Li, K. and Liu, D. and Varshavsky, A. Vis-a-vis: Privacy-preserving online social networking via virtual individual servers. In: Communication Systems and Networks (COMSNETS), 2011 Third International Conference on. IEEEp.1–10.

[37] Backes, M. and Maffei, M. A security API for distributed social networks. Network and Distributed System Security. 2011;.

[38] T. Hardjono, Ed. User-Managed Access (UMA) Core Protocol, draft-hardjono-oauth-umacore-05C; 2012. Available from: http://docs.kantarainitiative.org/uma/draft-uma-core.html , last access May 2014.

[39] Graffi, K. and Groß, C. and Stingl, D. and Hartung, D. and Kovacevic, A. and Steinmetz, R. LifeSocial.KOM: A Secure and P2P-based Solution for Online Social Networks. In: Proceedings of the IEEE Consumer Communications and Networking Conference. IEEE Computer Society Press; 2011. .

[40] Lucas, M. M. and Borisov, N. FlyByNight: mitigating the privacy risks of social networking. In: Proceedings of the 7th ACM Wks. on Privacy in the electronic society. WPES '08. ACMp.1–8.

[41] Zhu, F. and Lv, Q. ACEAC: A Novel Access Control Model for Cooperative Editing with Workflow. In: Electronic Commerce and Security, 2008 International Symposium onp.1010 –1014.

[42] FOAF Team. *FOAF* project; 2000. . http://www.foaf-project.org/ , last access May 2014.

[43] Lalas, E. and Papathanasiou, A. and Lambrinoudakis, C. Privacy and Traceability in Social Networking Sites. In: Informatics (PCI), 2012 16th Panhellenic Conference onp.127–132.

[44] Park, J. Usage Control: A Unified Framework for Next Generation Access Control. George Mason University; 2003.

[45] Samarati, P. and Capitani di Vimercati, S. Access control: Policies, models, and mechanisms. In: Foundations of Security Analysis and Design. Springerp.137–196.

[46] Capitani di Vimercati, S. and Foresti, S. and Samarati, P. Authorization and access control. In: Security, Privacy, and Trust in Modern Data Management. Springerp.39–53.

[47] Ferrari, E. Access control in data management systems. Morgan & Claypool; 2010.

[48] Hu, H. and Ahn, G-J. and Jorgensen, J. Detecting and resolving privacy conflicts for collaborative data sharing in online social networks. In: Proceedings of the 27th Annual Computer Security Applications Conference. ACSAC '11. ACMp.103–112.

[49] ITU-T-team. ITU-T Recommendation X.812. Data networks and open system communications security.; 1995. http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf , last access May 2014.

[50] Yuan, E. and Tong, J. Attributed based access control (ABAC) for web services. In: Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on. IEEE; 2005. .

[51] Goyal, V. and Pandey, O. and Sahai, A. and Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on Computer and communications security. ACMp.89–98.

[52] Wang, L. and Wijesekera, D. and Jajodia, S. A logic-based framework for attribute based access control. In: Proceedings of the 2004 ACM workshop on Formal methods in security engineering. ACMp.45–55.

[53] Kuhn, D. R. and Coyne, E. J. and Weil, T. R. Adding attributes to role-based access control. Computer. 2010;43(6):79–81.

[54] Tapiador, A. and Carrera, D. and Salvacha, J. Tie-RBAC: An application of RBAC to Social Networks. CoRR. 2012;.

[55] Munckhof, C. W. D. Content Based Access Control in Social Network Sites. Eindhoven University Of Technology; 2011.

[56] Villegas, W. and Ali, B. and Maheswaran, M. An Access Control Scheme for Protecting Personal Data. In: Proceedings of the 2008 Sixth Annual Conference on Privacy, Security and Trust. PST '08p.24–35.

[57] Ali, B. and Villegas, W. and Maheswaran, M. A trust based approach for protecting user data in social networks. In: Proceedings of the 2007 conference of the center for advanced studies on Collaborative research. CASCON '07. IBM Corp.p.288–293.

[58] Carminati, B. and Ferrari, E. and Heatherly, R. and Kantarcioglu, M. and Thuraisingham, B. A semantic web based framework for social network access control. In: Proceedings of the 14th ACM symposium on Access control models and technologies. SACMAT '09. ACMp.177–186.

[59] Carminati, B. and Ferrari, E. and Heatherly, R. and Kantarcioglu, M. and Thuraisingham, B. Semantic web-based social network access control. computers & security. 2011;30(2):108–115.

[60] Aiello, L. M. and Ruffo, G. LotusNet: Tunable privacy for distributed online social network services. Comput Commun. 2012;35(1):75–88.

[61] Ackermann, M. and Hymon, K. and Ludwig, B. and Wilhelm, K. Helloworld: An open source, distributed and secure social network. In: W3C Workshop on the Future of Social Networking; 2009. .

[62] Guha, S. and Tang, K. and Francis, P. NOYB: privacy in online social networks. In: Proceedings of the first Wks. on Online social networks. WOSN '08. ACMp.49–54.

[63] Luo, W. and Xie, Q. and Hengartner, U. FaceCloak: An Architecture for User Privacy on Social Networking Sites. 2009 International Conference on Computational Science and Engineering. p.26–33.

[64] Besenyei, T. and Földes, Á. M. and Gulyás, G. G. and Imre, S. StegoWeb: Towards the Ideal Private Web Content Publishing Tool. In: SECURWARE 2011, The Fifth International Conference on Emerging Security Information, Systems and Technologiesp.109–114.

[65] Kourtellis, N. and Finnis, J. and Anderson, P. and Blackburn, J. and Borcea, C. and Iamnitchi, A. Prometheus : User−Controlled P2P Social Data Management for Socially−Aware Applications. Ifip International Federation For Information Processing. p.212–231.

[66] Zhu, Y. and Hu, Z. and Wang, H. and Hu, H. and Ahn, G−J. A Collaborative Framework for Privacy Protection in Online Social Networks. Organization. p.1–15.

[67] Cutillo, L. A. and Molva, R. and Strufe, T. Safebook: Feasibility of transitive cooperation for privacy on a decentralized social network. 2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops. 2009;(217141):1–6.

[68] Frikken, K. B. and Srinivas, P. Key−allocation schemes for private social networks. In: Proceedings of the 8th ACM Wks. on Privacy in the electronic society. WPES '09. ACMp.11–20.

[69] Bethencourt, J. and Sahai, A. and Waters, B. Ciphertext-policy attribute-based encryption. In: Security and Privacy, 2007. SP'07. IEEE Symposium on. IEEEp.321–334.

[70] Baden, R. and Bender, A. and Spring, N. and Bhattacharjee, B. and Starin, D. Persona: an online social network with user−defined privacy. SIGCOMM Comput Commun Rev. 2009;39:135–146.

[71] Jahid, S. and Mittal, P. and Borisov, N. EASiER: encryption−based access control in social networks with efficient revocation. In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security. ASIACCS '11. ACMp.411–415.

[72] Jahid, S. and Nilizadeh, S. and Mittal, P. and Borisov, N. and Kapadia, A. DECENT: A decentralized architecture for enforcing privacy in online social networks. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on. IEEEp.326–332.

[73] Braghin, S. and Iovino, V. and Persiano, G. and Trombetta, A. Secure and Policy-Private Resource Sharing in an Online Social Network. In: Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom). IEEEp.872–875.

[74] Joe, M. Identity−based cryptography. IOS Press; 2009.

[75] Schlegel, R. and Wong, D. Private friends on a social networking site operated by an overly curious SNP. Network and System Security. p.430–444.

[76] Hohenberger, S. and Waters, B. Attribute-based encryption with fast decryption. In: Public-Key Cryptography–PKC 2013. Springerp.162–179.

[77] Hoyle, M. P. and Mitchell, C. J. On solutions to the key escrow problem. In: State of the Art in Applied Cryptography. Springerp.277–306.

[78] Gates, C. Access control requirements for web 2.0 security and privacy. IEEE Web. 2007;2(0).

[79] Tripunitara, Mahesh V. and Li, N. A theory for comparing the expressive power of access control models. J Comput Secur. 2007;15(2):231–272.

[80] Anwar, M. and Fong, P. W. L. and Yang, X. and Hamilton, H. Visualizing privacy implications of access control policies in social network systems. In: Proceedings of the 4th international workshop, and Second international conference on Data Privacy Management and Autonomous Spontaneous Security. DPM'09/SETOP'09. Springer-Verlagp.106–120.

[81] Villegas, W. A trust-based access control scheme for social networks. McGill University; 2008.

[82] Squicciarini, A. C. and Shehab, M. and Paci, F. Collective privacy management in social networks. In: Proceedings of the 18th international conference on World wide web. WWW '09p.521–530.

[83] Cheng, Y. and Park, J. and Sandhu, R. Relationship-Based Access Control for Online Social Networks: Beyond User-to-User Relationships. In: SocialComp.646–655.

[84] Simpson, A. On the need for user-defined fine-grained access control policies for social networking applications. In: Proceedings of the workshop on Security in Opportunistic and SOCial networks. SOSOC '08. ACMp.1:1–1:8.

[85] Besmer, A. and Lipford, H. R. and Shehab, M. and Cheek, G. Social applications: exploring a more secure framework. In: Proceedings of the 5th Symposium on Usable Privacy and Security. SOUPS '09. ACMp.2:1–2:10.

[86] Tootoonchian, A. and Saroiu, S. and Wolman, A. Lockr : Better Privacy for Social Networks. Design. 2009;.

[87] Buchegger, S. and Schiöberg, D. and Vu, L–H. and Datta, A. PeerSoN: P2P social networking: early experiences and insights. p.46–52.

[88] Zhang, H. and Wu, W. and Li, Z. Open Social based group access control framework for e-Science data infrastructure. In: E-Science (e-Science), 2012 IEEE 8th International Conference on. IEEEp.1–8.

[89] Ahmad, A. and Whitworth, B. Distributed access control for social networks. In: Information Assurance and Security (IAS), 2011 7th International Conference onp.68 –73.

[90] Kruk, S. and Grzonkowski, S. and Gzella, A. and Woroniecki, T. and Choi, H. D–FOAF: Distributed Identity Management with Access Rights Delegation. In: The Semantic Web ? ASWC 2006. Lecture Notes in Computer Science. Springerp.140–154.

[91] Wang, H. and Sun, L. Trust-Involved Access Control in Collaborative Open Social Networks. In: Proceedings of the 2010 Fourth International Conference on Network and System Security. NSS '10p.239–246.

[92] Seong, S–W. and Seo, J. and Nasielski, M. and Sengupta, D. and Hangal, S. and Teh, S. K. and Chu, R. and Dodson, B. and Lam, M. S. PrPl: a decentralized social networking infrastructure. p.8:1–8:8.

[93] Park, J. and Sandhu, R. A Position Paper: A Usage Control (UCON) Model for Social Networks Privacy. 2000;.

[94] Abdessalem, T. and Dhia, I. B. A reachability-based access control model for online social networks. In: Databases and Social Networks. DBSocial '11. ACMp.31–36.

[95] Dhia, I. B. Access control in social networks: a reachability-based approach. In: Proceedings of the 2012 Joint EDBT/ICDT Workshops. ACMp.227–232.

[96] Dhia, B. I. and Abdessalem, T. and Sozio, M. Primates: a privacy management system for social networks. In: Proceedings of the 21st ACM international conference on Information and knowledge management. ACMp.2746–2748.

[97] Fong, P. W. L. and Anwar, M. and Zhao, Z. A privacy preservation model for facebook-style social network systems. In: Proceedings of the 14th European conference on Research in computer security. ESORICS'09. Springer-Verlagp.303–320.

[98] Bruns, G. and Fong, P. WL. and Siahaan, I. and Huth, M. Relationship-based access control: its expression and enforcement through hybrid logic. In: Proceedings of the second ACM conference on Data and Application Security and Privacy. ACMp.117–124.

[99] Alizadeh, M. and Javadi, S. A. and Amini, M. and Jalili, R. Policy specification and enforcement in online social networks using MKNF+. In: Information Security and Cryptology (ISCISC), 2012 9th International ISC Conference on. IEEEp.48–53.

[100] Conti, M. and Hasani, A. and Crispo, B. Virtual private social networks. In: Proceedings of the first ACM conference on Data and application security and privacy. CODASPY '11. ACMp.39–50.

[101] Squicciarini, A. C. and Sundareswaran, S. Web-traveler policies for images on social networks. World wide web. 2009;12(4):461–484.

[102] Bertino, E. and Catania, B. and Ferrari, E. and Perlasca, P. A logical framework for reasoning about access control models. In: Proceedings of the sixth ACM symposium on Access control models and technologies. SACMAT '01. ACMp.41–52.

[103] Ganta, S. Expressive power of access control models based on propagation of rights. George Mason University; 1996.

[104] Chander, A. and Mitchell, J. C. and Dean, D. A State-Transition Model of Trust Management and Access Control. In: Proceedings of the 14th IEEE workshop on Computer Security Foundations. CSFW '01. Washington, DC, USA: IEEE Computer Societyp.27–.

[105] Habib, L. and Jaume, M. and Morisset, C. Formal definition and comparison of access control models. Journal of Information Assurance and Security. 2009;4:372–378.

[106] Salim, F. and Reid, J. and Dawson, E. An administrative model for UCON$_{ABC}$. In: Proceedings of the Eighth Australasian Conference on Information Security. AISC '10p.32–38.

[107] Sandhu, R. and Park, J. Usage control: A vision for next generation access control. Computer Network Security. p.17–31.

[108] Dewan, P. and Shen, H. Flexible meta access-control for collaborative applications. In: Proceedings of the 1998 ACM conference on Computer supported cooperative work. CSCW '98. ACMp.247–256.

[109] Sandhu, R. S. and Coyne, E. J. and Feinstein, H. L. and Youman, C. E. Role-based access control models. Computer. 1996;29(2):38 –47.

[110] Sandhu, R. and Bhamidipati, V. and Munawer, Q. The ARBAC97 model for role-based administration of roles. ACM Trans Inf Syst Secur. 1999;2(1):105–135.

[111] Kim, D. and Solomon, M. G. Fundamentals of Information Systems Security. Jones & Bartlett; 2010.

[112] Thompson, M. R. and Essiari, A. and Mudumbai, S. Certificate-based authorization policy in a PKI environment. ACM Trans Inf Syst Secur. 2003;6(4):566–588.

[113] Squicciarini, A. C. and Shehab, M. and Wede, J. Privacy policies for shared content in social network sites. The VLDB Journal. 2010;19(6):777–796.

[114] Ahmad, A. and Whitworth, B. and Janczewski, L. More Choices, More Control: Extending Access Control by Meta-rights Reallocation. In: Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference onp.1113 –1118.

[115] Jung, Y. and Joshi, J. BD. CPBAC: Property-based Access Control Model for Secure Cooperation in Online Social Networks. Computers & Security. 2013;.

[116] Ren, Y. Access control in a cooperative editing system. In: Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference onp.77 –80.

[117] Prilla, M. and Ritterskamp, C. Collaboration support by co-ownership of documents. In: Proceedings of the 2006 conference on Cooperative Systems Design: Seamless Integration of Artifacts and Conversations – Enhanced Concepts of Infrastructure for Communicationp.255–269.

[118] Imine, A. and Cherif, A. and Rusinowitch, M. A Flexible Access Control Model for Distributed Collaborative Editors. In: Proceedings of the 6th VLDB Workshop on Secure Data Management. SDM '09p.89–106.

[119] Lorch, M. and Adams, D. B. and Kafura, D. and Koneni, M. S. R. and Rathi, A. and Shah, S. The PRIMA System for Privilege Management, Authorization

and Enforcement in Grid Environments. In: Proceedings of the 4th International Workshop on Grid Computing. GRID '03. IEEE Computer Societyp.109–.

[120] Wedde, H. F. and Lischka, M. Cooperative role-based administration. In: Proceedings of the eighth ACM symposium on Access control models and technologies. ACMp.21–32.

[121] Sandhu, R. and Krishnan, R. and Niu, J. and Winsborough, W. Group-centric models for secure and agile information sharing. Computer Network Security. p.55–69.

[122] Sandhu, R. and Bijon, K. Z. and Jin, X. and Krishnan, R. RT-based administrative models for community cyber security information sharing. In: CollaborateComp.473–478.

[123] Edwards, W. K. Policies and roles in collaborative applications. In: Proceedings of the 1996 ACM conference on Computer supported cooperative work. CSCW '96. ACMp.11–20.

[124] Sikkel, K. A group-based authorization model for cooperative systems. In: Proceedings of the fifth conference on European Conference on Computer-Supported Cooperative Work. ECSCW'97. Kluwer Academic Publishersp.345–360.

[125] Zhang, Z. Y. and Huang, T. and Wu, Q. T. and Pu, J. X. A CSCW-Enabling Integrated Access Control Model and its Application. Key Engineering Materials. 2011;460:96–105.

[126] Thomas, R. K. Team-based access control (TMAC): a primitive for applying role-based access controls in collaborative environments. In: Proceedings of the second ACM workshop on Role-based access control. RBAC '97. ACMp.13–19.

[127] Cohen, E. and Thomas, R. K. and Winsborough, W. H. and Shands, D. Models for coalition-based access control (CBAC). In: SACMATp.97–106.

[128] Gligor, V. and Khurana, H. and Koleva, R. and Bharadwaj, V. and Baras, J. On the negotiation of access control policies. In: Security Protocols. Springerp.188–201.

[129] Jin, J. and Ahn, G-J. Role-based access management for ad-hoc collaborative sharing. In: Proceedings of the eleventh ACM symposium on Access control models and technologies. SACMAT '06. ACMp.200–209.

[130] Carminati, B. and Ferrari, E. Collaborative access control in on-line social networks. In: Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference onp.231 –240.

[131] Xiao, Q. and Tan, K-L. Peer-aware collaborative access control in social networks. In: Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on. IEEEp.30–39.

[132] Besmer, A. and Richter Lipford, H. Moving beyond untagging: photo privacy in a tagged world. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '10. ACMp.1563–1572.

[133] Wishart, R. and Corapi, D. and Marinovic, S. and Sloman, M. Collaborative Privacy Policy Authoring in a Social Networking Context. In: Policies for Distributed Systems and Networks (POLICY), 2010 IEEE International Symposium onp.1 –8.

[134] Sun, Y. and Zhang, C. and Pang, J. and Alcade, B. and Mauw, S. A trust-augmented voting scheme for collaborative privacy management. In: Proceedings of the 6th international conference on Security and trust management. STM'10. Springer-Verlagp.132–146.

[135] Lin, D. and Rao, P. and Bertino, E. and Li, N. and Lobo, J. Policy decomposition for collaborative access control. In: Proceedings of the 13th ACM symposium on Access control models and technologies. SACMAT '08. ACMp.103–112.

[136] Merlo, A. and Armando, A. Cooperative access control for the Grid. In: Information Assurance and Security (IAS), 2010 Sixth International Conference onp.228 –233.

[137] Braghin, S. and Ferrari, E. and Trombetta, A. Combining access control and trust negotiations in an On-line Social Network. In: Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference onp.1 –10.

[138] Radatz, J. and Geraci, A. and Katki, F. IEEE standard glossary of software engineering terminology. IEEE Std. 1990;610121990:121990.

[139] Dawson, F. and Mansour, S. and Silverberg, S. RFC-2445: Internet Calendaring and Scheduling Core Object Specification (iCalendar). 1998;.

[140] Yeung, C. A. and Liccardi, I. and Lu, K. and Seneviratne, O. and Berners-Lee, T. Decentralization: The future of online social networking. In: W3C Workshop on the Future of Social Networking Position Papers; 2009. .

[141] Suri, P. and Garg, N. Software Reuse Metrics: Measuring Component Independence and its Applicability in Software Reuse. International Journal of Computer Science and Network Security. 2009;9(5):237–248.

[142] Breslin, J. and Decker, S. The Future of Social Networks on the Internet: The Need for Semantics. IEEE Internet Computing. 2007 Nov;11(6):86–90.

[143] Bellamy-McIntyre, J. and Luterroth, C. and Weber, G. OpenID and the Enterprise: A Model-Based Analysis of Single Sign-On Authentication. In: Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International. IEEEp.129–138.

[144] OAuth Working Group. The OAuth 2.0 Authorization Framework; 2011. http://tools.ietf.org/html/rfc6749, last access May 2014.

[145] Leiba, B. Oauth web authorization protocol. Internet Computing, IEEE. 2012;16(1):74–77.

[146] Brickley, D. and Miller, L. FOAF Vocabulary Specification 0.98; 2010. http://xmlns.com/foaf/spec/#term_workplaceHomepage , last access May 2014.

[147] Harary, F. and Norman, R. Z. Graph theory as a mathematical model in social science. University of Michigan; 1953.

[148] Carminati, B. and Ferrari, E. and Perego, A. Private Relationships in Social Networks. In: Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Wks. IEEE Computer Societyp.163–171.

[149] Covington, M. J. and Sastry, M. R. A contextual attribute-based access control model. In: Proceedings of the 2006 international conference on On the Move to Meaningful Internet Systems: AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET - Volume Part II. OTM'06p.1996–2006.

[150] Grompanopoulos, C. and Gouglidis, A. and Mavridis, I. A Use-based Approach for Enhancing UCON. 2013;.

[151] Gross, J. and Yellen, J. Graph theory and its applications. CRC Press, Inc.; 1999.

[152] Scowen, Roger S. Extended BNF-a generic base standard. Technical report, ISO/IEC 14977. http://www. cl. cam. ac. uk/mgk25/iso-14977. pdf , last access May 2014; 1998.

[153] Zhang, L. and Tu, W. Six Degrees of Separation in Online Society. In: Proceedings of the WebSci'09: Society On-Linep.87 –90.

[154] NIST. American National Standard for Information TechnologyRole Based Access Control; 2003.

[155] OWASP-team. OWASP Top 10; 2013. http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf , last access May 2014.

[156] Attrapadung, N. and Imai, H. Attribute-based encryption supporting direct/indirect revocation modes. In: Cryptography and Coding. Springerp.278–300.

[157] European-Comission. Data Protection Directive (95/46/EC) of the European Parliament and of the council; 2013. http://ec.europa.eu/justice/policies/privacy/docs/95-46-ce/dir1995-46_part1_en.pdf , last access May 2014.

[158] Lipford, H. R. and Hull, G. and Latulipe, C. and Besmer, A. and Watson, J. Visible Flows: Contextual Integrity and the Design of Privacy Mechanisms on Social Network Sites. In: CSE (4)p.985–989.

[159] H. Saltzer, J. and D. Schroeder, M. The Protection of Information in Computer Systems. The Protection of Information in Computer Systems. 1975;(9):1278–1308.

[160] Borcea-Pfitzmann, K. and Pfitzmann, A. and Berg, M. Privacy 3.0 := Data Minimization + User Control + Contextual Integrity. Information Technology. 2011;53(1):34–40.

[161] U. S. Department of commerce, William M. Daley, Secretary National Institute of standars and technology. Name of Standard: Entity Authentication Using Public Key Cryptography. 1997;.

[162] NIST Internet Time Service; 2010. http://www.nist.gov/pml/div688/grp40/its.cfm , last access May 2014.

[163] Calhoun, C. Imagined Communities and Indirect Relationships: Large Scale Social Integration and the Transformation of Everyday Life. 1991;.

[164] Jammalamadaka, R. C. and Gamboni, R. and Mehrotra, S. and Seamons, K. and Venkatasubramanian, N. iDataGuard: An interoperable security middleware for untrusted internet data storage. In: Proceedings of the ACM/IFIP/USENIX Middleware '08 Conference Companion. Companion '08. ACMp.36–41.

[165] Capitani di Vimercati, S. and Foresti, S, and Jajodia, S and Paraboschi, S and Samarati, P. A data outsourcing architecture combining cryptography and access control. In: Proceedings of the 2007 ACM workshop on Computer security architecture. CSAW '07. ACMp.63–69.

[166] Google Team. Google: Privacy Policy; 2012. http://www.google.cz/intl/en/policies/privacy/ , last access May 2014.

[167] Graffi, K. and Gross, C. and Mukherjee, P. and Kovacevic, A. and Steinmetz, R. LifeSocial.KOM: A P2P-Based Platform for Secure Online Social Networks. In: Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference onp.1 –2.

[168] Salomaa, A. Public–key cryptography. Springer; 1996.

[169] Maurer, U. M. and Yacobi, Y. A Non–interactive Public–Key Distribution System. Des Codes Cryptography. 1996;9(3):305–316.

[170] Kleinberg, J. The small-world phenomenon: An algorithmic perspective. In: Proceedings of the thirty-second annual ACM symposium on Theory of computing. ACMp.163–170.

[171] Adamic, L. and Adar, E. How to search a social network. Social Networks. 2005;27(3):187–203.

[172] Kwak, H. and Lee, C. and Park, H. and Moon, S. What is Twitter, a social network or a news media? In: Proceedings of the 19th international conference on World wide web. ACMp.591–600.

[173] Ugander, J. and Karrer, B. and Backstrom, L. and Marlow, C. The anatomy of the facebook social graph. arXiv preprint arXiv:11114503. 2011;.

[174] Nah, F. F-H. A study on tolerable waiting time: how long are web users willing to wait? Behaviour & Information Technology. 2004;23(3):153–163.

[175] Crampton, J. and Khambhammettu, H. Delegation in role-based access control. In: Computer Security–ESORICS 2006. Springerp.174–191.

[176] Zhang, Z. and Yang, L. and Pei, Q. and Ma, J. Research on Usage Control model with delegation characteristics based on OM-AM methodology. In: Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on. IEEEp.238–243.

[177] Lunt, C. and Abrams, J. and Sanchez, S. Method of inducing content uploads in a social network-United States Patent (7117254); 2006. http://www.google.com/patents/US7117254 , last access May 2014.

[178] Mazurek, M. L. and others. Access Control for Home Data Sharing: Attitudes, Needs and Practices. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '10. ACMp.645–654.

[179] Olson, J. S. and Grudin, J. and Horvitz, E. A Study of Preferences for Sharing and Privacy. In: CHI '05 Extended Abstracts on Human Factors in Computing Systems. CHI EA '05. ACMp.1985–1988.

[180] He, J. and Chu, W. W. A social network-based recommender system (SNRS). Springer; 2010.

[181] Li, N. and Zhang, N. and Das, S. K. Preserving relation privacy in online social network data. Internet Computing, IEEE. 2011;15(3):35–42.

[182] Kang, J. and Shilton, K. and Estrin, D. and Burke, J. and Hansen,M. Self-Surveillance Privacy. Datenschutz und Datensicherheit - DuD. 2011;35(9):624–628.

[183] Park, J. and Sandhu, R. and Cheng, Y. A User–Activity–Centric Framework for Access Control in Online Social Networks. Internet Computing, IEEE. 2011;15(5):62 –65.

[184] Harman, G. Reasoning, Meaning, and Mind. Oxford University Press; 1999.

[185] Capitani di Vimercati, S. and Foresti, S. and Samarati, P. Authorization and Access Control. Security, Privacy, and Trust in Modern Data Management. p.39–53.

[186] Zhang, X. and Park, J. and Parisi-Presicce, F. and Sandhu, R. A logical specification for usage control. In: Proceedings of the ninth ACM symposium on Access control models and technologies. SACMAT '04. ACMp.1–10.

Appendix A

Acronyms and abbreviations

The following tables present acronyms and abbreviations listed in alphabetical order:

Acronym	**Term**
A	Authorizations.
ABAC	Attribute Based Access Control.
ABE	Attribute Based Encryption.
ACM	Access Control Model.
ACP	Access Control Policies in a system.
ADF	Access Control Decision Function.
AEF	Access Control Enforcement Function.
AM	Authorization Manager
AO	Administrative Objects.
AR	Administrative Rights.
ATT	Attributes.
AU	Authorizing User.
CA	Certification Authority.
CP-ABE	Ciphertext-Policy Attribute-Based Encryption.
D_k	Decryption with key k.
DAC	Discretionary Access Control.
DB	Data Base.
DoS	Denial of Service.
E	Direct relationships.
E_k	Encryption with key k.
FOAF	Friend-Of-A-Friend.
HTML	HyperText Markup Language.
IBE	Identity Based Encryption.
IdP	Identity Provider.
KP-ABE	Key-Policy Attribute-Based Encryption.
MAC	Mandatory Access Control.
NIST	National Institute of Standards and Technology.
O	Objects.
OBAC	Ontology Based Access Control.
PDP	Policy Decision Point.

Acronym	Term
PEP	Policy Enforcement Point.
P2P	Peer to Peer.
TBAC	Trust Based Access Control.
TTP	Trusted Third Party.
TW	Temporal Workload.
R	Use Rights.
RBAC	Role Based Access Control.
RelBAC	Relationship Based Access Control.
RP	Requesting Party.
RT	Relationships.
S	Subjects.
UDF	Usage Decision Facility.
UEF	Usage Enforcement Facility.
UMA	User-Managed Access.
URL	Uniform Resource Locator.
WBSN	Web Based Social Network.
XML	eXtensible Markup Language.

Appendix B

Publications

This Section lists the references of published works that have been derived during the realization of this doctoral thesis. The preliminary work over which this thesis has been based are also introduced here.

Published

- Based on the thesis:

1. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. "U+F Social Network Protocol: Achieving interoperability and reusability between Web Based Social Networks". 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, 2012.

2. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, B. Ramos. "Control de Acceso en Redes Sociales Web". XII Reunión española sobre Criptología y Seguridad de la Información (RECSI), 2012.

3. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. Security and Privacy Preserving in Social Networks. Chapter "User-Managed Access Control in Web Based Social Networks". In Security and Privacy Preserving in Social Networks (pp. 97-137). Springer Vienna.

4. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. "Seguridad en Redes Sociales: problemas, tendencias y retos futuros", VII Congreso Iberoamericano en Seguridad Informática (CIBSI), 2013.

5. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. "SoNeUCON$_{ABC}$, an expressive usage control model for Web-Based Social Networks", Journal of Computers Security, 2014.

6. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. "Extended U+F Social Network Protocol: Interoperability, reusability, data protection and indirect relationships in Web Based Social Networks", Journal of Systems and Software, 2014.

7. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. "$SoNeUCON_{ADM}$: the administrative model for $SoNeUCON_{ABC}$ usage control model", XIII Reunión española sobre Criptología y Seguridad de la Información (RECSI), 2014.

8. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. "CooPeD: Co-owned Personal Data Management", Journal of Computers Security, 2014.

- Related to the thesis:

9 L. González-Manzano, B. Brost, M.Aumuller. "An architecture for trusted PaaS cloud computing for personal data", Proceedings of the Workshop Wissenschaftliche Ergebnisse der Trusted Cloud Initiative, Springer Verlang, 2014.

Submitted

- Related to the thesis:

10 E. Palomar, L. González-Manzano, A. Alcaide, A. Galán. "Implementing a Privacy-enhanced ABC System for Online Social Networks with Co-Ownership Management", Journal of Computer Networks.

11 L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. "Towards a trade-off between privacy and rewards in Web-Based Social Networks advertising", Information Systems Journal.

12 L. González-Manzano, J. M de Fuentes, Ana I. González-Tablas, A. Ribagorda. "VAADapt - Adapting VAAD for an enriched and privacy-preserving advertisement dissemination in VANETs", Information Sciences.

Appendix C

Expressive power analysis of ACMs for WBSNs

The appendix presents the analysis of the expressive power of ACMs for WBSNs in terms of the definition of policies presented in Section 2.5.2. First, inducting reasoning is applied to generalize the policies (Section C.1). Afterwards, a set of 24 ACMs for WBSNs are analysed (Section C.2).

C.1 Generalizing access control policies

In order to determine the expressive power of ACMs, analysing that a concrete policy, related to a feature, can be expressed by a specific model and not by another one is not enough. The chief question that comes up is, whether it is possible to assert that any model that expresses a policy associated with a feature is able to define any other policy related to this feature. Inductive reasoning can be applied to address this issue and reach a general rule after having reasoned from a set of cases [184]. By applying inductive reasoning it is established that if a particular policy (P), based on a concrete feature (F), can be expressed by a specific model (M), then to generalize that M can express any kind of P based on F is possible. In other words, using this technique it is generalized the number of P that can be created in a particular M.

In this work inductive reasoning is applied to features *distance* (F1), *common-contacts* (F2), *clique* (F3) and *multi-path* (F4). On the contrary, it cannot be ap-

plied to *direction* (F5) and it would be extremely tedious in respect to *fine-grained* (F6) due to several reasons. Regarding F5, this feature exclusively requires the creation of directional and bidirectional relationships and consequently, generalization does not have to be applied. By contrast, according to F6, the amount of attributes that can be managed is extremely assorted and their generalization is unattainable.

Table C.1: Inductive reasoning

Distance (F1)

$(C1) A \xrightarrow{D_a} B \ \wedge \ B \xrightarrow{D_b} C \ \wedge C \xrightarrow{D_c} D \ \Rightarrow \ D \longrightarrow DAT(A)$

$(C2) A \xrightarrow{D_a} B \ \wedge \ B \xrightarrow{D_b} C \ \wedge C \xrightarrow{D_c} E \ \wedge E \xrightarrow{D_c} D \ \Rightarrow \ D \longrightarrow DAT(A)$

$(G) X \xrightarrow{D_x} Y_1 \ \wedge \ Y_1 \xrightarrow{D_{y1}} Y_2 \ \wedge \ ... \ \Rightarrow \ Y_{n+1} \longrightarrow DAT(X)$

Given that this feature can be generalized for $n \geqslant 3$ where n corresponds to the number of hops, P1 is proposed assuming $n = 3$.

Common contacts (F2)

$(C1) A \xrightarrow{D_{a1}} B \ \wedge \ C \xrightarrow{D_{a1}} B \ \Rightarrow \ C \longrightarrow DAT(A)$

$(C2) A \xrightarrow{D_{a1}} B \ \wedge \ C \xrightarrow{D_{a1}} B \ \wedge \ A \xrightarrow{D_{a1}} D \ \wedge \ C \xrightarrow{D_{a1}} D \ \Rightarrow \ C \longrightarrow DAT(A)$

$(C3) A \xrightarrow{D_{a1}} B \ \wedge \ C \xrightarrow{D_{a1}} B \ \wedge \ A \xrightarrow{D_{a1}} D \ \wedge \ C \xrightarrow{D_{a1}} D \ \wedge \ A \xrightarrow{D_{a1}} E \ \wedge \ B \xrightarrow{D_{a1}} E \ \Rightarrow \ C \longrightarrow DAT(A)$

$X \xrightarrow{D_{x1}} Y_1 \ \wedge \ ... \wedge \ X \xrightarrow{D_{xn}} Y_n \ \wedge \ Z \xrightarrow{D_{x1}} Y_1 \ \wedge \ ... \wedge \ Z \xrightarrow{D_{xn}} Y_n \ \Rightarrow \ Z \longrightarrow DAT(X)$

Given that this feature can be generalized for $n \geqslant 1$ where n corresponds to the number of common contacts, P2 is proposed assuming $n = 3$.

Clique (F3)

$(C1) A \longleftrightarrow B \ \wedge \ A \longleftrightarrow C \ \wedge \ B \longleftrightarrow C \ \Rightarrow \ A \longrightarrow DAT(B, \ C) \ \wedge \ B \longrightarrow DAT(A, \ C) \ \wedge \ C \longrightarrow DAT(A, \ B)$

$(C2) A \longleftrightarrow B \ \wedge \ A \longleftrightarrow C \ \wedge \ A \longleftrightarrow D \ \wedge \ B \longleftrightarrow D \ \wedge \ B \longleftrightarrow C \ \wedge \ C \longleftrightarrow D \ \Rightarrow \ A \longrightarrow DAT(B, \ C, \ D)$
$\wedge \ B \longrightarrow DAT(A, \ C, \ D) \ \wedge \ C \longrightarrow DAT(A, \ B, \ D) \ \wedge \ D \longrightarrow DAT(A, \ B, \ C)$

$(G) X \longleftrightarrow Y_1 ... \ \wedge \ X \longleftrightarrow Y_n \ \wedge \ Y_1 \longleftrightarrow Y_2 ... \ \wedge \ Y_1 \longleftrightarrow Y_n \ \wedge \ ... \ \wedge \ Y_{n-1} \longleftrightarrow Y_n \ \Rightarrow \ X \longrightarrow DAT(Y_1, ..., Y_n)$
$\wedge \ Y_1 \longrightarrow DAT(X, Y_2, ..., Y_n) ... \ \wedge \ Y_n \longrightarrow DAT(X, Y_1, ..., Y_{n-1})$

Given that this feature can be generalized for $n \geqslant 2$ where $n + 1$ corresponds to the number of contacts in the clique, P3 is proposed assuming $n = 2$.

Multi-path (F4)

$(C1) A \xrightarrow{D/I_{b1}} B \ \wedge \ A \xrightarrow{D/I_{b2}} B \ \wedge \ b1 \neq b2 \ \Rightarrow \ B \longrightarrow DAT(A)$

$(C2) A \xrightarrow{D/I_{b1}} B \ \wedge \ A \xrightarrow{D/I_{b2}} B \ \wedge \ A \xrightarrow{D/I_{b3}} B \ \wedge \ b1 \neq b2 \ \wedge \ b1 \neq b3 \ \wedge \ b2 \neq b3 \ \Rightarrow \ B \longrightarrow DAT(A)$

$(G) X \xrightarrow{D/I_{y1}} Y \ \wedge \ ... X \xrightarrow{D/I_{yn}} Y \ \wedge \ y1 \neq ....yn \ \Rightarrow \ Y \longrightarrow DAT(X)$

Given that this feature can be generalized for $n \geqslant 2$ where n corresponds to the number of different paths that connect the administrator and the requester, P4 is proposed assuming $n = 2$.

Applied predicates

- X is direct or indirectly connected with Y by the relationship z: $X \xrightarrow{D/I_z} Y$
- X accesses to data, DAT, of contact Y: $X \longrightarrow DAT(Y)$
- X accesses to data, DAT, of contacts $\{Y_1, Y_2, ..., Y_n\}$: $X \longrightarrow DAT(Y_1, Y_2, ..., Y_n)$
- X and Y are bidirectional contacts: $X \longleftrightarrow Y$

Table C.1 presents the application of inductive reasoning to F1, F2, F3 and F4. For each feature, a set of cases CZ (being Z $\in$ ℵ) is established to reach a general rule (G), considering that both, CZ and G, describe particular and general access control policies respectively. An access control policy is structured following the pattern $X \Rightarrow Y$, where X refers to the set of conditions to satisfy in order that Y happens, that is, certain rights are granted to the requester by the administrator. Cases are incremental - the higher cases include the previous ones. In other words, given C1, C2 and C3, expressing C3 means that C2 and C1 are satisfactorily expressed too because they are included in C3. Then, it is expected, given the general nature of ACMs, that ACMs that express C3 are expressive enough to define any policy from G.

Regarding feature F1 *distance*, the proposed policy for case C1 establishes that access is granted to users (D) who are at a distance of three hops from the administrator (A). Similarly, C2 presents a policy which grants access to users (D) who are at a distance of four hops from the administrator (A). Then, following this reasoning, the specification of C1 and C2 can be generalized as an access control policy (G) that involves users connected by a number n of hops, $n \geqslant 3$. Then, P1, the policy that will be used to analyse the expressive power of ACMs, is proposed considering $n = 3$.

In what concerns feature F2 *common-contacts*, the proposed policy for case C1 corresponds to a policy in which access is granted to users (C) who have a contact (B) in common with the administrator (A). Similarly, C2 corresponds to a policy which grants access to users (B) who have a pair of contacts (B and D) in common with the administrator (A). Following an analogous reasoning, in the general case G access is granted to users who have a number n of common contacts with the administrator. Thus, P2, the policy that will be used to analyse the expressive power of ACMs, is proposed considering $n = 3$.

On the other hand, according to feature F3 *clique*, case C1 presents a policy

that grants access to users that belong to a clique composed of three users (A, B and C, where A is the administrator and B or C the requester). Analogously, C2 corresponds to a policy that grants access to four users involved in a clique (A, B, C and D, where A is the administrator and B, C or D the requester). Therefore, generalized case G refers to a policy that grants access to users who belong to a clique composed of $n+1$ users, being n the number of users involved different from the administrator with $n \geqslant 2$. Specifically, it refers to the establishment of as many bidirectional relationships as existing edges in a complete graph composed of $n+1$ nodes and, consequently, the number of created bidirectional relationships are $\frac{(n-1)\times n}{2}$. Then, P3, the policy that will be used to analyse the expressive power of ACMs, is proposed assuming $n = 2$.

Finally, in relation to feature F4 *multi-path*, case C1 presents a policy that grants access to users (B) with whom the administrator (A) is directly or indirectly connected by a pair of different paths. Likewise, C2 refers to a policy that grants access to users (B) that are connected to the administrator (A) by three different paths. Following such reasoning, generalized case G presents a policy that grants access to users connected to the administrator by a number n of different paths. Therefore, P4, the policy that will be used to analyse the expressive power of ACMs, is proposed assuming $n = 2$.

C.2 Analysis of the expressive power of ACMs for WBSNs

In order to evaluate the expressive power of ACMs for WBSNs, a total of 24 proposals of ACMs suitable for that context have been selected and if the policy language linked to each ACM is expressive enough to specify the set of access control policies defined in Section 2.5.2 has been evaluated.

In the following Sections, the 24 proposals classified as RBAC, RelBAC, ABAC, TBAC and OBAC models, are briefly introduced and the results of the evaluation

are presented. In some cases the specification of policies includes the creation of additional elements, such as predicates, attributes or relationship types, which are created according to each proposal specifications. Besides, some policies cannot be specified because the model has not got the appropriate level of expressive power. In particular, partially specified policies are marked with parenthesis, such as (Pi) and the policies that cannot be specified are not mentioned.

C.2.1 Role based access control (RBAC) models

[ACM 1] Role based access control for Social Networks [24] This approach bases on relationship management. Relationships are represented as the connection between a pair of users taking each of them one specific role.

A WBSN involves the management of multiple elements, for instance user attributes. Nonetheless, this model exclusively includes relationship management offering quite restrictive procedures. Relationship types are the only relationship property that the model considers.

Before defining access control policies, a set of roles and permissions have to be specified:

- *Roles* R={friend}
- *Permissions* P={read}
- $R \times P \Rightarrow$ the defined permission is assigned to all roles

According to [24], access control policies are identified as social relations ($SR(s)$) that a particular user (s) owns. They consist of three elements, {User1, User2, <User1's role, User2's role>}, where User1 and User2 corresponds to the couple of users involved in the relationship. Assuming that a is the administrator and s the requester, the set of access control policies proposed in Section 2.5.2 are defined as follows:

P5 $SR(a) = \{a,\ s,\ < friend,\ friend >\}$

$SR(s) = \{s,\ a,\ < friend,\ friend >\}$

P6 $SR(a) = \{a,\ s,\ < friend,\ friend >\}$

Due to the simplicity of the model just a couple of policies can be specified. Regarding P5, it is satisfactorily defined through a pair of social relations. It is assumed, but not explicitly mentioned, that an access control policy can be composed of more that a single SR. Analogously, P6 is properly defined.

[ACM 2] Tie-RBAC [54] A relation is defined as a set of ties of the same type between senders and receivers. Each relation involves the sender's assignation of the receiver to a role with permissions. Besides, ties are non-reciprocal, that is, they are unidirectional and having a tie with a particular user does not imply the existence of a tie on the other way round.

The specification of policies is quite limited. Concerning access control policies proposed in Section 2.5.2, just a pair of them can be created. The administrator is a, s is the requester and the tie used is "Friend" which has the read permission linked to an object:

P5 $Tie_1 = Friend\ from\ a\ to\ s$ and $Tie_2 = Friend\ from\ s\ to\ a$

P6 $Tie_1 = Friend\ from\ a\ to\ s$

Therefore, P5 and P6 are satisfactorily expressed.

[ACM 3] Distributed access control [89] Looking for the decentralization of access control in WBSNs, this model bases on sharing resources regarding relationship type or closeness. Besides, roles (called Local Roles, LR), permissions (called Attestation Certificates, AC) and objects that belong to users (called Namespace, NS) are managed.

This model manages users, relationships and objects but not in a fine-grained way because just a single policy from Section 2.5.2, P6, can be specified.

The following elements are required:

- LR={friend}
- AC={r is granted to all roles in LR}
- NS={o and other data the administrator has}

In what concerns previous elements, considering i a particular domain (e.g. a particular set of photos), τ a set of security labels attached to objects and OC an object group (e.g. privacy labels), policy P6, called conditions (con), can be specified as follows:

P6 $con(s,o,i) = isactive(s,\ NS) \wedge hasLR(s,\ i) \wedge hasOC(o,\ i,\ \tau) \wedge hasAC(AC,\ LR)$

C.2.2 Trust based access control (TBAC) models

[ACM 4] Social access control (SAC) [57] SAC bases access control management on exploiting trust relationships between users. Policies are established regarding the trust placed in users (τ) and the confidentiality attached to each object ($t_c(o)$), such that if $\tau > t_c(o)$ the access is granted and denied otherwise.

SAC allows the definition of policies P4 and P5, but partially:

(P4) $\tau = high$

(P5) $\tau = high$

In this model trust is placed in users instead of relationships. Nonetheless, it is assumed that if a user gets access to an object with a certain kind of trust attached, it can be compared with the establishment of a trust relationship with this user. As a result, P4 is defined to some extent. In addition, relationships are inherently bidirectional and then, P5 is implicitly defined.

[ACM 5] Rule based access control [28] Carminati *et al.* propose a rule-based access control model to allow users the specification of access rules for their contents. Policies are expressed as constraints on the type, depth, and trust level of existing relationships. Moreover, certificates are applied to attest the authenticity of the relationships.

Relationships are unidirectional and, even considering a bidirectional relationship as a pair of unidirectional ones, provided specifications do not detail how to manage bidirectional relationships in access control policies.

In order to define access control policies proposed in Section 2.5.2, the following relationships types are required:

- *Relationship Types*={FriendOf, RelativeOf}

The last type of the pair has been created to meet the goals of the analysis presented herein. Types are non-fixed and can be deliberately defined in the approach. According to the model, access control rules are composed of the *oid* which is the identifier of the requester resource and a set of predicates. Each predicate is composed of four relationship elements: {Node, Relationship Type, Jumps, Trust level}, where "Node" refers to the administrator (a), that is, the owner of the requested data.

Policies are defined as follows:

(P1) $(oid,\ \{(a, RelativeOf, 3, *)\})$

(P4) $(oid,\ \{(a, FriendOf, 1, 10)\})$

P6 $(oid,\ \{(a, FriendOf, 1, *)\})$

Only three policies can be specified and just P6 is satisfactorily defined. On the one hand, regarding P1, the indirect relationship is not completely expressed. An indirect relationship composed of three hops can be defined but neither different roles in each hop nor particular relationship preferences can be pointed out. On

the other hand, P4 expresses the fact that the relationship is completely trusted (considering a level of trust from 0 to 10) but it does not specify multiple paths.

[ACM 6] Reachability-based access control model [94] This model expresses access control rules as reachability constraints which encode the path between the requester and the administrator of the requested object. More specifically, similar to [28], this model bases on the specification of access control policies regarding the trust and distance of WBSN users.

Regarding elements managed in a WBSN, this model focuses on relationships and users. Nonetheless, relationships management is quite limited. Trust and distance are the only managed attributes and there are not tools to deal with features such as common friends, cliques or multiple paths (used in P2, P3 and P4 respectively).

In this analysis, the following relationships and user properties, all of them described in [94], are applied:

- *Relationship*={Friend}
- *User properties*={age, gender, studies}

Specifically, access control policies consist of a tuple of three elements, {Requested object, Relationships path, Trust threshold}. Furthermore, "Relationships path" consists, at the same time, of four elements, {Starting user, Relationship (with a '+' or a '-' referring '+' to outgoing relationships and '-' to incoming), Relationship depth, User properties}. Also, "Trust threshold" is 0 when it is not considered in a particular rule.

The set of policies from Section 2.5.2 that can be specified in this model, is constructed as follows, considering that a is the administrator:

(P1) $(o,\ \{(a,\ Friend^{+}[1,2,3])\},\ 0)$

(P4) $(o,\ \{(a,\ Friend^{+}[1])\},\ 10)$

(P5) $(o,\ \{(a,\ Friend^{+}[1]),\ (o, Friend^{-}[1])\},\ 0)$

P6 $(o,\ \{(a,\ Friend^{+}[1])\},\ 0)$

(P7) $(o,\ \{(a,\ Friend^{+}[1][age = 30][gender = female])\},\ 0)$
$(o,\ \{(a,\ Friend^{+}[1][age = 40][gender = female][studies = c.science])\},\ 0)$
$(o,\ \{(a,\ Friend^{+}[1][studies = c.science])\},\ 0)$

A great set of policies can be expressed using this model, though completely successful results are not reached. Indeed, P6 is the only policy properly specified. By contrast, P1 is defined to some extent. It grants access to users, located at three hops without pointing out relationship preferences. Similarly, P4 specifies the existence of a highly trusted relationship but without specifying multiple paths. Likewise, differing from the proposed P5, the created policy is simultaneously satisfied by unidirectional and bidirectional relationships. Similarly, P7 is not satisfactorily specified and it requires the definition of a set of three policies. In particular, the only operator applied in the specification of attributes values is "=" and thus, the definition of granting access to users under a certain age is unattainable. Analogously, according to the model specifications, attributes are uni-valued and it cannot be defined granting access to users who have multiple degrees, e.g. a degree in computer science and other in physics.

[ACM 7] Personal Data Access Control (PDAC) [56] This model manages access by computing the trusted distance (d_{trust}) between users. A particular level of trust is linked to each user and each user is located at a certain distance in the social network graph. Subsequently, once data is requested, d_{trust} is calculated regarding the trust and distance of the requester to the administrator.

Specifically, assuming that a is the administrator and o the requested object, policies can be identified as the establishment of a trust interval such that (*accept limit*, *reject limit*). The *accept limit* ($C_a(a, d)$) refers to the largest trusted distance that will be considered to grant the access and the *reject limit* corresponds to the

smallest trusted distance ($C_r(a, d)$). As a result, access is granted if the calculated d_{trust} is within the established interval. It is considered that trust distance for a *friend* is 1, for a *friend-of-friend* is 2 and for *friend-of-friend-of-friend* is 3 and so on. Intermediate trust levels are also managed and, for instance, 0.8 refers to a very good friend and 0.9 to a good friend.

Concerning above specifications, the set of access control policies proposed in Section 2.5.2 that can be defined in this ACM is the following one:

(P1) (3,3)

(P4) (0,0.5)

P6 (0,1)

PDAC achieves the complete definition of P6, establishing that access is granted if requesters are friends with any level of trust between 0 and 1, that is, if they are good, very good, extremely good friends, etc. Nonetheless, P1 and P4 are partially expressed. P1 defines that only those users who are located at a distance three from the administrator get access, leaving aside the specification of relationship types and the relationship creation time. Likewise, P4 specifies granting access to users who are highly trusted but the definition of multiple paths is not achieved.

[ACM 8] Trust in Collaborative Open Social Networks [91] Wang *et al.* present a fine-grained access control scheme for WBSNs that manages access control through a purpose-based approach. Data is linked to a set of purposes that form a hierarchy and can change dynamically.

Access control policies are constructed as rules composed of seven elements: *Data* which identifies the requested data; *Sub* that refers to requesters to whom the access is granted; *RelT* which corresponds to the type of the relationship between the requester and the data owner; *Purp* that corresponds to the right that users request; *Dmax* which corresponds to the maximal relationship depth; *Tmin* that

refers to the minimal trust; and *Obli* that refers to requirements that have to be satisfied before granting access. As a result, a policy rule is defined as: (*Data, Sub, RelT, Purp, Dmax, Tmin, Obli*).

Considering that o is the requested object, r is the requester, relationship types are $friend$, $relative$ and $neighbour$, $read$ is the purpose, trust rates from 0 to 10 and there are not obligations ($\varnothing$), the definition of policies from Section 2.5.2 is the following one:

(P1) (o, s, $relative$, $read$, 3, 5, $\varnothing$)

(P4) (o, s, $friend$, $read$, 1, 10, $\varnothing$)

P6 (o, s, $friend$, $read$, 1, 5, $\varnothing$)

This approach allows the definition of three policies but just P6 is completely defined. P1 is partially defined as roles at each distance can be different and the specification of relationship attributes is possible. Similarly, P4 is partially defined because this proposal only allows the establishment of high trust relationships but not multiple paths.

C.2.3 Relationship based access control (RelBAC) models

[ACM 9] Privacy preservation model [97] This model bases on the generalization of Facebook access control mechanism. It demonstrates that the expression of several policies not currently supported by Facebook can be carried out, such as the sharing of data between common friends or friends involved in a clique.

In this ACM, users and relationships are the main managed elements. Similar to previous proposals, it does not deal with attributes management and policies such as P7 cannot be defined. Furthermore, as it is based on Facebook, the establishment of unidirectional relationships like the one proposed in P6 is unreachable.

Assuming that u and v refer to a pair of WBSN users, s is the requester, G is the complete social network represented as a graph and γ refers to a particular

communication state that describes the communication history between the administrator (a) and s, the set of access control policies proposed in Section 2.5.2 that can be defined by this model is the following one:

(P1) $only\text{-}me = \gamma(u,\ a,\ G,\ \gamma).u = a$

$only\text{-}friends = only\text{-}me\ \vee\ (\gamma(u,\ v,\ G,\ \gamma).\{u,\ v\} \in E(G))$

$friends\text{-}of\text{-}friends = only\text{-}friends\ \vee\ (\gamma(v,\ s,\ G,\ \gamma).(\exists v' \in Sub\{s,\ v'\} \in E(G)\ \wedge\ \{v',\ v\} \in E(G)))$

P2 $common - friends_3 = only\text{-}friends\ \vee\ (\gamma(s,\ a,\ G,\ \gamma).|N_G(s)\ \cap\ N_G(a)| \geqslant 3)$

P3 $clique_3 = only\text{-}friends \vee (\gamma(s,\ a,\ G,\ \gamma).(\exists G'.G' \subseteq G \wedge G'\ \cong\ K_3 \wedge \{s,\ a\} \subseteq V(G')))$

(P4) $path_2 = friends\text{-}of\text{-}friends\ \wedge\ friends\text{-}of\text{-}friends = (only\text{-}friends = only\text{-}me\ \vee\ (\gamma(u,\ v,\ G,\ \gamma).\{u,\ v\} \in E(G)))\ \wedge\ (only\text{-}friends = only\text{-}me\ \vee\ (\gamma(z,\ x,\ G,\ \gamma).\{z,\ x\} \in E(G)))$

P5 $only\text{-}friends = only\text{-}me\ \vee\ (\gamma(u,\ v,\ G,\ \gamma).\{u,\ v\} \in E(G))$

Due to the similarity with Facebook, results are quite interesting. Regarding P1, it is partially expressed because indirect relationships have a maximum depth of two and fine-grained management of relationships is not considered either. An interesting policy is P4 which, considering the possibilities offered by this model, is defined through the combination of existing access control policies. Then, even not currently supported by Facebook, the establishment of policies that involve multiple paths is supported. However, contrary to the proposed P4, the created one requires the specification of each path length and also, the definition of being different paths is missing. By contrast, P2, P3 and P5 are satisfactorily defined.

[ACM 10] Relation based access control [27, 19] This model focuses on capturing the idea that an authorization decision, a policy, bases exclusively on the relationship between the administrator and the requester. The main issue runs

towards the specification of policies capable of expressing WBSN features such as having common friends or the establishment of cliques.

This ACM, as its name suggests, focuses on relationship without considering attributes management, either relationships, objects or users attributes. Then, P7, which is particularly focused on user attributes, cannot be defined.

Regarding policies from Section 2.5.2, the following set of relationships identifiers are applied:

- I={friend, neighbour, relative}

Considering that the administrator (a) is identified by the prepositional symbol $@p$ [19], v and u refer to WBSN users, and s refers to the requester, access control policies are defined as follows:

(P1) $(@p.\langle relative\rangle(\neg u \wedge \neg v \wedge \langle neighbour\rangle v)(\neg u \wedge \neg v \wedge \neg s \wedge \langle friend\rangle s))$

P2 $s \vee \langle friend\rangle s \vee ((\langle friend\rangle\langle friend\rangle\langle friend\rangle s) \oplus (\langle friend\rangle\langle friend\rangle\langle friend\rangle s))$

P3 $s \vee (\neg s \wedge \langle friend\rangle s \wedge @p.\langle friend\rangle(\neg p \wedge \neg s \wedge \langle friend\rangle s))$

(P4) $(\langle friend\rangle(\neg s \wedge \langle neighbour\rangle s)) \wedge (\langle friend\rangle(\neg v \wedge \langle neighbour\rangle v))$

P5 $\langle friend\rangle s \wedge \langle -friend\rangle s$

P6 $\langle friend\rangle s$

This model is one of the most expressive models attaining the successful definition of four policies, P2, P3, P5 and P6. However, in respect to P1, given that this model is not focused on attributes management, the specification of the proposed relationship creation time is infeasible. On the other hand, according to P4, it specifies the existence of a pair of different paths of two hops.

[ACM 11] Relation based access control through hybrid logic [98] This proposal presents a RelBAC model that uses hybrid logic to express access control policies. Indeed, as mentioned above, this is one of the first contributions which particularly mention and work to achieve expressive power. The model is similar to the one proposed in [19] but applying other type of logic.

In general, binary relationships are managed, tagged with a set of labels in I, being S the set of principals, subjects, involved in them. Moreover, it is distinguished *own* to refer to the owner, the administrator, and *req* to refer to the requester, and the symbol @ is used to define a policy in regard to both principals. Specifically, labels applied herein are the following ones:

- I={friend, neighbour, relative}

As a result, policies from Section 2.5.2 are defined as follows:

(P1) $@_{own}\langle relative\rangle((req \wedge \langle neighbour\rangle req) \wedge req \wedge \langle friend\rangle req$

P2 $@_{own}(req \vee \langle friend\rangle req \wedge \langle friend\rangle_3 \langle friend\rangle req$

(P4) $@_{own}(\langle friend\rangle req \wedge \langle neighbour\rangle req$

P5 $@_{own}\langle friend\rangle req \wedge @_{req}\langle friend\rangle own$

P6 $@_{own}\langle friend\rangle req$

This analysis shows the possibilities offered by hybrid logic. Nevertheless, the difficult task of expressing cliques is pointed out [98], as well as it is just briefly mentioned the management of user attributes. These attributes are applied in terms of types (called labels). They are quite similar to relationships attributes and thus, P1 is partially expressed. Similarly, P4 defines a pair of different paths but their length has to be pre-defined. On the contrary, P2, P5 and P6 are successfully defined.

[ACM 12] User relationship-based access control (UURAC) model [83] UURAC is an online social network model focused on existing WBSN relationships and the establishment of policies through regular expressions. Its main challenge goes towards the definition of policies that involve direct and indirect relationships with different types in each hop.

This novel model specially bases on direct and indirect relationships management. Consequently, P1, P5 and P6 are the policies that can be specified by UURAC.

In this ACM, $\sum$ corresponds to the set of managed relationship types and *action* refers to the set of actions that can be requested:

- $\sum$=$\{f, n, r\}$ where f corresponds to friend, n to neighbour and r to relative.
- *action*=$\{rd\}$ where rd refers to the read permission.

This model proposes a non-fixed set of $\sum$. Thus, relationships types neighbour n and relative r have been created. Furthermore, there are different types of policies in UURAC. In this regard, as the analysis performed herein focuses on policies specified by an administrator in regard to a resource (o), policies called *target resources policies* are the ones applied. These policies consist of three elements <*action*, *resource*, (*starting node*, *path rule*)>, where *path rule* corresponds to a set of predicates connected by disjunctions and conjunctions. Besides, each predicate is composed of (*relationship path*, *hopcount*) where *hopcount* refers to the maximum number of edges on the relationship path.

The set of policies from Section 2.5.2 that can be defined in this ACM is the following one:

(P1) $(r, o, (r*, 1) \wedge (n*, 2) \wedge (f*, 3))$

P5 $(r, o, (f*, 1))$

Alluding to this model's name, relationships are the main managed elements. On the one hand, P1 is partially defined because the lack of attributes management

prevents from detailing the proposed relationship creation time. On the other hand, P5 is properly defined. Indeed, UURAC explicitly mentions that relationships are unidirectional and their bidirectional nature exists simultaneously.

[ACM 13] Multiparty Access Control (MPAC) for Online Social Networks [32] MPAC focuses on capturing multiparty authorization requirements. It makes possible the collaborative management of shared data in WBSNs.

Access control policies are composed of five elements: *controller*, who is the user who manages access control; *ctype*, that refers to the type of the controller; *accessor*, who is the user to whom the access is granted and it may consist of the user name, the relationship type and the group name; *data*, which corresponds to the identifier of the requested data and the level of data sensitivity; and *effect* refers to the permission granted, that is, permit or deny. Therefore, assuming that the administrator a permits access to an object o with a sensitivity level sl, policies from Section 2.5.2 are defined in this ACM as follows:

(P1) $< a,\ OW,\ \{< friend-of-friend,\ RN >\}, < o,\ sl >,\ permit >$

P5 $< a,\ OW,\ \{< friend-of,\ RN >\}, < o,\ sl >,\ permit >$

Applying this model just a couple of policies can be defined, being P5 the only one completely specified. Also, it should be noticed that relationships have a maximum length of two and they are inherently considered bidirectional.

[ACM 14] A reachability-based approach [95] This novel ACM bases on connection characteristics between WBSN users. It tries to generalize access control policies in terms of users properties, indirect relationships and complex relationship composed of direct relationships of different types.

Access control policies, called access rules, consist of the tuple (rid, ACS) where rid is the identifier of the requested resource and ACS refers to the set of access conditions (ac) to satisfy. Besides, each ac is composed of (o, p) where o is the

starting node, that is, the resource administrator, and p refers to a path of ordered steps. Each step is also composed of four elements (r, dir, I, C) where r is the type of the relationships, dir is the orientation of the relationship edge ($+, -$ or $*$ in case of bidirectionality), I is the set of authorized distances and C the set of conditions regarding user properties. Then, the existence of following elements is assumed:

- Relationship types= {Relative, Neighbour, Friend}
- User properties= {age, gender, trust}

As a result, assuming that the administrator is a and the requested resource is ro, the set of access control policies proposed in Section 2.5.2 are defined as follows:

(P1) $(ro, (a, (Relative, ^+, 1), (Neighbour, ^+, 1), (Friend, ^+, 1))))$

(P4) $(ro, (a, (Friend, ^+, 1, (trust = 1))))$

P5 $(ro, (a, (Friend, ^*, 1)))$

P6 $(ro, (a, (Friend, ^+, 1)))$

(P7) $(ro, (a, (Friend, ^+, 1, (gender = female, age < 30))))$

This ACM is significantly expressive as it allows the partial specification of five policies from the set proposed in Section 2.5.2. P1 is partially defined since the relationship creation time is not specified. Similarly, P4 is defined to some extent. A trust relationship is specified but not the existence of multiple paths. Likewise, even being possible the definition of user attributes, disjunctions cannot be specified in P7. By contrast, unidirectional and bidirectional relationships are appropriately expressed.

[ACM 15] Primates [96] The ACM proposed in this approach is quite similar to the one presented in [95]. Access control is managed through reachability constraints based on paths between WBSNs users and user properties.

Concerning policies, called access rules, they consist of four elements such that (u, r, P, C) where u is the resource owner, r the requester resource, P the path and C the set of constraints on the attributes of the requester. Besides P consists of constraints on the path that connects the resource owner and the requester, and each constraint is, simultaneously, composed of the tuple (l, dir, I) where l is the type of the relationships, dir the direction of the relationships ($\rightarrow$, $\leftarrow$ or $\Leftrightarrow$) and I the minimum and maximum depth of the path. Therefore, assuming the same relationship types and user attributes as those defined in [95] and considering a to be the administrator and o the requested object, policies from Section 2.5.2 are defined in this proposal as follows:

(P1) $(a, o, (\text{`}Relative', \rightarrow, (1,1)), (\text{`}Neighbour', \rightarrow, (1,1)), (\text{`}Friend', \rightarrow, (1,1)), -)$

(P4) $(a, o, (\text{`}Friend', \rightarrow, (1,1)), [trust = 1])$

P5 $(a, o, (\text{`}Friend', \Leftrightarrow, (1,1)), -)$

P6 $(a, o, (\text{`}Friend', \rightarrow, (1,1)), -)$

(P7) $(a, o, (\text{`}Friend', \rightarrow, (1,1)), [gender = female, age < 30])$

Due to the similarity with [95], conclusions are equivalent. The definition of P5 and P6 is complete and the definition of P1, P4 and P7 is partial.

C.2.4 Attribute based access control (ABAC) models

[ACM 16] $UCON_{ABC}$ for social networks [93] $UCON_{ABC}$ for social networks bases on $UCON_{ABC}$ [44, 106] usage control model. It is developed under the perspective of Attribute Based Access Control (ABAC) models [185]. The definition of policies bases on subjects, objects and the environment, as well as subjects and objects attributes (ATT(S) and ATT(O) respectively). Besides, this model can be used to model MAC, DAC and RBAC access control policies, as well as certain authorization processes of Digital Rights Management (DRM).

P3 and P4 cannot be defined because they base on complex relationships management that is is not included in this model. Furthermore, it is noticeable that access control policies are defined by data owners, referred to as administrators. Then, in this model, according to [106], it is assumed that the administrator, who is considered a subject, is the user who administrates requested objects and thus, manages ATT(O).

Specifically, access control focuses on the specification and management of predicates that express relationships between subjects and objects [186, 106]. According to access control policies proposed in Section 2.5.2, the following ATT(S), ATT(O) and predicates are defined. Notice that except for the predicates *permit* and *in*, which are presented in [186], the rest of predicates and attributes have been created herein following the model specifications.

Attributes:

- $ATT(S)$={Age, Gender, Studies, Friends, Neighbours, Relatives} where Friends, Neighbours and Relatives are the lists of friends, neighbours and relatives, respectively, that the user has. Moreover, each list is composed of a set of attributes:
  - Friends={$\{User1_{Id}, relationshipTrust, ...\}$
  - Neighbours={$\{User1_{Id}, relationshipTrust, ...\}$
  - Relatives={$\{User1_{Id}, relationshipTrust, ...\}$
- $ATT(O)$={$Object1_{Id}, ...$}

Predicates:

- permit($s \in S,\ o \in O,\ r$): it grants access permission (r) over an object (o) to a particular subject (s).

- in($s \in S$, $Friends/Neighbours/Relatives\ of\ v \in S$): it returns the existence of not of a friendship/neighbour/relative relationship between a pair of users (s and v). Notice that s has to be within the list of friends/neighbours/relatives of v.

- commonFriends($Friends\ of\ s \in S$, $Friends\ of\ v \in S$, n): it returns a positive or negative value regarding if the list of friends of a subject (s) has n subjects in common with the list of friends of another subject (v), being $n \in \aleph$.

Policies have been constructed following the process described in [186]. Moreover, the administrator of the requested object o is referred to as a and s corresponds to the requester. Given that in this work policies are inherently unidirectional, access control policies established by a are described below:

(P1) $in(a, s.Neighbours) \land a.Neighbours[s].creationTime < 2001 \longrightarrow permit(s, a.o, r)$

P2 $commonFriends(a.Friends, s.Friends,\ 3) \longrightarrow permit(s, a.o, r)$

P5 $in(a, s.Friends) \land in(s, a.Friends) \longrightarrow permit(s, a.o, r)$

P6 $in(s, a.Friends) \longrightarrow permit(s, a.o, r)$

P7 $(s.gender = female \land s.age < 30 \lor (s.gender = female \land s.age < 40 \land s.studies = \{C.Science\}) \lor (s.studies = \{C.Science\} \land s.studies = \{Physics\}) \longrightarrow permit(s, a.o, r)$

A great set of access control policies from Section 2.5.2 are satisfactorily defined. Conversely, in respect to P1, the relationship creation time is specified but the proposed indirect relationship (related to P3 and P4) is not because, as mentioned above, this model does not focus on relationships management. By contrast, the rest of policies are satisfactorily expressed.

[ACM 17] ACON: Activity-Centric Access Control for Social Computing [30] ACON focuses on managing sessions and activities and it is also developed on the bases of $UCON_{ABC}$ [44, 106] usage control model. Thus, it allows the definition the same policies as $UCON_{ABC}$ for social networks, namely, P2, P5, P6 and P7 and P1 to some extent.

[ACM 18] Content-based access control for social networks [55] This proposal presents an automatic ACM that selects a particular policy for a post of an added message according to its content. Then, the main characteristic of this ACM is the identification of the content of each particular object.

Concerning policies, they are composed of five elements: *priority* which corresponds to the relevance of the policy; *name* which refers to an unique identifier; *explanation* that points out how the system has concluded; *attributes* which refer to the list of managed attributes; and *rules*, that indicate how elements match in the destination profile. In sum, they are expressed as:

Priority — Name — Explanation — Attributes — rules

Supposing that *priority* is p, *name* is id and *explanation* refers to the description of each policy $expPX$ with $X\ in\{1-7\}$, the set of access control policies proposed in Section 2.5.2 are defined as follows:

P6 p — id — $expP6$ — - — $is\text{-}friend$

P7 p — id — $expP6$ — $-gender$, $-age$, $-studies$ — $[gender(female)AND(age < 30)]\ OR$
$[gender(female)AND(age < 40)ANDstudies(C.Science)]\ OR\ [gender(female)AND studies(C.Science)ANDstudies(Physics)]$

Therefore, a pair policies, P6 and P7, are successfully expressed.

[ACM 19] Persona [70] This proposal bases on Attribute Based Encryption (ABE) cryptography and consequently, the ACM which lays the bases of this work

is ABAC. Specifically, *Persona* applies CP-ABE (recall Section 2.3). Thus, users creates keys regarding a set of attributes, encrypt data using encryption keys and distribute decryption keys among their contacts.

The strength of this approach focuses on dealing with untrusted service storages. Nevertheless, the specification of expressive policies is not one of the main goals of [70]. In this work, "friend" is the only attribute used within policies, though disjunctive and conjunctive operators can be applied. Consequently, access control policies proposed in Section 2.5.2 are specified in this proposal as follows:

P6 friend

The set of access control policies that Persona allows to create is not flexible enough. Policy elements are limited to attributes connected by disjunctive and conjunctive operators, that is, it can be compared with the management of groups. Moreover, the necessity of delivering decryption keys to chosen users supportss the unidirectional nature of relationships and thus, P6 is properly defined.

[ACM 20] EASiER [71] Similar to *Persona [70]*, this proposal focuses on ABE and, specially on CP-ABE. Therefore, policies are constructed through attributes combined with disjunctive and conjunctive operators.

According to policies from Section 2.5.2, the attribute applied is "friend" and they are defined as follows:

P6 friend

This approach, as *Persona*, does not focus on the establishment of expressive policies. Besides, assuming that decryption keys are delivered from data owners to the requester, established relationships are unidirectional and P6 is satisfactorily defined.

[ACM 21] Secure and Policy-Private Resource Sharing [73] This proposal presents an ABE solution, thereby based on an ABAC model, that achieves the defi-

nition of expressive policies regarding the social network graph (users represented as nodes and edges as relationships). Specially, Distance-Based Revokable Attribute Encryption (DBRA) is applied. Links are established between users to exchange decryption keys and the specification of access control policies bases on resource attributes and the distance between the resource owner and the administrator.

Regarding access control policies, they are composed of a set of access rules (ar) composed of conditions ($cond$), such that $\langle ar_1 \rangle, \langle ar_2 \rangle, ..., \langle ar_n \rangle$ where $ar = cond_1, cond_2, ...cond_m$. A particular condition is $dist(u, d)$ being u the requester and d the maximum distance between the requester and the administrator. Assuming the existence of resource attributes "relatives", "neighbours" and "friends", the set of policies from Section 2.5.2 that can be defined in this proposal is the following one:

(P1) {⟨RelativeType= "relatives".$dist(u, 1)$⟩, ⟨NeighbourType= "neighbours".$dist(u, 2)$⟩, ⟨FriendType= "friends".$dist(u, 3)$⟩}

P6 {⟨FriendType= "friends".$dist(u, 3)$⟩}

A relevant point of this approach is the management of resource attributes. It allows the definition of the policies P1 and P6. In what concerns P1, the indirect relationship is defined to some extent because the relationship creation time cannot be managed. Conversely, P6 is properly defined following the same bases as in *Persona* [70] and *EASier* [71].

C.2.5 Ontology based access control (OBAC)models

[ACM 22] An Ontology-based Access Control Model for Social Networking Systems (OSNAC) [29] OSNAC focuses on the management of a semantic ontology for WBSNs. It captures the WBSN semantic and constructs a model to manage it. In particular, it is described as a rule-based access control policy model in which rules are specified at user and at system level. The former refers to per-

sonal authorization rules established by users regarding protected resources and the latter corresponds to rules that govern the overall privacy policy of the system.

This model focuses on managing users, data, relationships and user attributes. By contrast, relationship attributes are left aside and together with the restrictive possibilities for creating rules, access control is not managed in a fine-grained way.

To express user policies the following properties are required, where *sn* and *ac* allude to relationships and actions respectively:

- *Properties*={sn:isFriendOf, sn:isNeighbourOf, sn:isRelativeOf, sn:hasGender, sn:isYoungerThan, ac:canRead}

This model provides an interesting set of properties opened to the inclusion of new ones. Except for *isFriendOf* and *canRead*, presented properties have being created according to the model specifications. In fact, *isRelativeOf* and *isNeighbourOf* follow the same bases as *isFriendOf*.

Considering that v, u and t refer to WBSN users, s corresponds to the requester and a is the administrator access control policies proposed in Section 2.5.2 are constructed as follows:

(P1) $sn : isRelativeOf(a,\ u) \wedge sn : isNeighbourOf(u,\ v) \wedge sn : isFriendOf(v,\ s) \wedge ac : canRead(s,\ o)$

P2 $sn : isFriendOf(a,\ t) \wedge sn : isFriendOf(a,\ u) \wedge sn : isFriendOf(a,\ v) \wedge sn : isFriendOf(s,\ t) \wedge sn : isFriendOf(s,\ u) \wedge sn : isFriendOf(s,\ v) \wedge ac : canRead(s,\ o)$

P3 $sn : isFriendOf(a,\ s) \wedge sn : isFriendOf(a,\ u) \wedge sn : isFriendOf(s,\ a) \wedge sn : isFriendOf(s,\ u) \wedge sn : isFriendOf(u,\ a) \wedge sn : isFriendOf(u,\ s) \wedge ac : canRead(s,\ o)$

(P4) $sn : isFriendOf(a,\ v) \wedge sn : isFriendOf(v,\ s) \wedge sn : isFriendOf(a,\ u) \wedge sn : isFriendOf(u,\ s) \wedge ac : canRead(s,\ o)$

P5 $sn : isFriendOf(a,\ s)\ \wedge sn : isFriendOf(s,\ a)\ \wedge\ ac : canRead(s,\ o)$

P6 $sn : isFriendOf(a,\ s)\ \wedge\ [r \leftarrow\ ac : canRead(s,\ o)]$

(P7) $sn\ :\ hasGender(s,\ Female)\ \wedge\ sn\ :\ isYoungerThan(s,\ 30)\ \wedge\ ac\ :\ canRead(s,\ o)$

$sn\ :\ hasGender(s,\ Female)\ \wedge\ sn\ :\ isYoungerThan(s,\ 40)\ \wedge\ sn\ :\ hasStudied(s,\ C.Science)\ \wedge\ ac : canRead(s,\ o)$

$sn : hasStudied(s,\ C.Science)\ \wedge\ ac : canRead(s,\ o)$

Applying this model all policies from Section 2.5.2 can be defined to some extent. In particular, P2, P3, P5 and P6 are satisfactorily expressed. On the contrary, even defining the indirect relationship proposed in P1 and given the lack of relationship attributes management, the existence of a relationship established before 2,000 is not specified. Likewise, P4 is partially defined. Multiple paths can be established but all of them with a particular length. Thus, the P4 gives access to users connected to the administrator by a pair of paths of two hops. Furthermore, the fact that paths are different is unspecified too. Finally, P7 is quite successfully defined through the establishment of as many access control policies as conjunctions. Nevertheless, the model does not manage multi-valued properties and granting access to a user who has studied computer science and physics becomes infeasible.

[ACM 23] Semantic web based framework [58] The general idea is to define a WBSN in terms of an ontology based on users' profiles, resources, relationships between users and between users and resources. Using this ontology the social network is modelled as a Social Network Knowledge Base (SNKB). Specifically, three types of policies are distinguished: *authorization policies* that consist of granting users permissions to execute privileges on objects; *admin policies* that state users who may specify access control policies for a certain privilege on an object; and *filtering policies* that establish prohibitions. Relationships have a particular trust assigned to them and policies are established accordingly. Besides, relationships

are unidirectional and the bidirectional nature is created as a pair of unidirectional ones.

Concerning policies, SWRL language is used to specified access control policies. Nonetheless, it cannot be used to deal with bidirectional relationships and they have to be managed out of SWRL. In general, in SWRL, access control policies are represented as antecedents, that encode conditions included in policies, and consequents, that encode authorizations and prohibitions. Considering the ontology applied in this ACM, the following instances are applied:

- Instances: $Relative$, $Neighbour$, $Friend$ and $Data$

Assuming that the administrator a grants read access to an object o to the requester r, access control policies proposed in Section 2.5.2 are defined as follows:

(P1) $Read : Relative(a, ?targetSubject1) \wedge Neighbour(?targetSubject1, ?targetSubject2) \wedge$

$$Friend(?targetSubject2, ?targetSubject3) \wedge Data(?o) \Rightarrow Read(?r, ?o)$$

P6 $Read : Relative(a, ?targetSubject1) \wedge Data(?o) \Rightarrow Read(?r, ?o)$

As a result, a pair of policies can be defined, being P6 the only one completely specified. On the other hand, P1 lacks the definition of the duration of the relationship.

[ACM 24] Online social networks using MKNF+ [99] A prioritized ontology based on an ACM for protecting users' data is proposed in [99]. It consists of a Minimal Knowledge and Negation as Failure (MKNF) formalism that combines Decryption Logic (DL) and rules created by Answer Set Programming (ASP). Furthermore, this model includes priority as an access control policy element to prevent conflicts caused by contradictions between each user's access control policies.

Concerning policies, they are composed of two types of predicates, DL-predicates and non-DL-predicates. The former bases on DL language and the latter focuses on

unary or binary predicates. Specifically, the following predicates, already defined in [99], are the ones applied herein:

- DL-Relationships=$\{IS\text{–}FRIEND\text{–}OF(Person,\ Person),\ ELEMENT(Object)\}$ where $ELEMENT$ may refer to a photo, a message or any other element in a WBSN.
- Non-DL-Concepts=$\{o(Object),\ s(Person)\}$

Assuming that src refers to the requested resource, sbj corresponds to the requester, a refers to the data owner and p refers to a certain type of priority, the following policy is defined:

P6 $K\ (?src),\ K\ s(?sbj),\ K\ IS - FRIEND - OF(a,\ ?sbj),\ K\ ELEMENT(?src) \rightarrow K\ permit(a, ?sbj, READ, ?src, p)$

In sum, relationships are pointed out as directed label edges and then, P6 is properly defined.

Appendix D

SoNeUCON$_{ABC}$ enforcement functions

The notation used to define each function corresponds to the name of the function, the input parameters (arguments), a set of predicates that refers to the establishment of variables or conditions and the returned value if required. It is based on [154] and it is formally represented as follows: $Function - Name(Arguments) \triangleleft$ $Predicate1$ $Predicate2$ ... $[Return - Value] \triangleright$

Moreover, symbol . is used to access to the content of an element. For instance, given a user (s), $s.age$ is used to access to the user's age. Besides, the expression $list[pos]$ refers to the access to an element located in position pos within the list $list$. For example, given the list i $=\{v, t, y\}$, $i[1]$ corresponds to t. Finally, it shlould be noticed that functions $MatchC$ and $MatchO$, that refer to the verification of conditions and obligations respectively, have to be implemented according to each particular case.

In the following Sections the enforcement functions for $SoNeUCON_{ABC}$ (Section D.1) and the enforcement functions for the extension of $SoNeUCON_{ABC}$ to manage co-ownership (Section D.2) are described. They are alphabetically ordered by name.

D.1 Enforcement functions for $SoNeUCON_{ABC}$

The access control enforcement functions applied to $SoNeUCON_{ABC}$ are the following:

CalculateCliquePaths The goal of this function is to identify the number of paths of different lengths that the construction of a clique of a certain number of users (δ) requires. Positions of the returned list correspond to path lengths and values of the returned list refer to the number of paths of each length. A list is returned if the calculus can be performed and "null" otherwise.

$$CalculateCliquePaths(\delta); out\ result : ListINTEGER) \lhd$$
$$result = ((if(\delta = 2) \Rightarrow ListClique[0] = 1) \vee (if(\delta > 2)$$
$$\Rightarrow (\sum_{K=1}^{\delta} ListClique[K] = (P(K,N) + 1))))\rhd$$

ContinuityCheckAccess This function is rather similar to $CheckAccess$, being distinguished a pair of issues. First, rt is already computed and then, just policy elements have to be verified. Second, this function is called once attributes have changed or the usage process has concluded the evaluation of $\rho_{on-going}$.

$$ContinuityCheckAccess(s,o,r,\rho,\partial_b,\partial_c,rt; out\ result : BOOLEAN) \lhd \rho_s \in$$
$$\rho; \rho_o \in \rho; \rho_{rt} \in \rho; subAtt = GetSubAtt(s,\rho_s);$$
$$objAtt = getObjAtt(o,,\rho_o); a = GetAdmin(o)$$
$$result = (\forall\rho((if(\rho_s\ NOT\ \varnothing) \Rightarrow Match(subAtt,\rho_s)) \wedge (if(\rho_o\ NOT\ \varnothing) \Rightarrow$$
$$Match(objAtt,\rho_o)) \wedge (if(\rho_{rt}\ NOT\ \varnothing)$$
$$\Rightarrow MatchRT(\rho_{rt},rt)) \wedge r = \rho.r \wedge MatchB(s,o_1,r,\rho,\partial_b)$$
$$\wedge MatchC(s,o,r,\rho,\partial_c)))\rhd$$

CreateRT This is a recursive function that focuses on creating rt from the WBSN graph, G, given the administrator (a) and the requester (s) of a particular request. Departing from the administrator node a, the process starts by visiting each of the contacts of a [$GetNumContacts$/ $GetConnectedUser$]. When the algorithm visits node v, the contacts of v are also visited [$CreateRT$] recursively until

the length of the path between a and the node being currently visited reaches 6 or node s is found. Then, if the path length reaches 6, the algorithm continues visiting the remaining contacts of the previous node in the path if any. If node s is reached, then the corresponding enriched path is stored (with all its forward and backward relationships and their attributes).

$CreateRT(v, s, hop; out\ result : rt) \lhd$
$result = ((\forall hop < 6) \longrightarrow (nC = GetNumContacts(v) \wedge (\forall i < nC \longrightarrow (c =$
$getConnectedUser(v, i) \wedge (Store forward\ and\ backward$
$relationships\ \ and\ \ length) \wedge ((if(c = s) \Rightarrow (Path\_completed)) \vee$
$(if(c\ \ NOT\ \ v) \Rightarrow CreateRT(c, s, hop + 1)) \vee (if(c = v) \Rightarrow$
$(Path\_broken))))))\rhd$

FindSubjectPolicies This function returns policies defined by a particular user.

$FindSubjectPolicies(s; out\ result : P) \lhd$
$result = (policies\ of\ s)\rhd$

GetAdmin This function, taking as input an object (o), returns the user who is its administrator. If the administrator of o is not found, "null" is returned.

$GetAdmin(o; out\ result : SUBJECT) \lhd$
$result = (administrator\ of\ o)\rhd$

GetConditions/ GetObligations This function returns, if exits, conditions or obligations within a given policy.

$GetConditions/GetObligations(\rho; out\ result : partial_b/partial_c) \lhd$
$result = (conditions\ or\ obligations\ within\ \rho)\rhd$

GetConnectedUser This function returns the id of a user directly connected to a given one. It is considered that contacts are stored ordered and then, they are returned according to a given position (pos). An id is returned if exists or "null" otherwise.

$GetConnectedUser(s, pos\ result : STRING) \lhd$

$result = (id\ of\ a\ user\ in\ position,\ pos,\ connected\ to\ s) \rhd$

GetDirectRelAtt The goal of this function is, given a ψ which corresponds to the set of conditions of a path ($pathCond$), to create a list τ which consists of conditions of direct forward and backward relationships found at a particular position ($pLength$) of $pathCon$. The process bases on identifying the set of forward and backward relationships between a semicolon (";") found at $(pLength - 1)$ and a semicolon found at $pLength$. Recall that operator semicolon, ";", is applied to distinguish hops of a path.

$GetDirectRelAtt(pathCon, pLength); out\ result : ListE) \lhd$

$cont = 1$

$result = (\forall i < pathCon.length \longrightarrow ((if(pathCon[i] = ";" \wedge$

$cont = pLength) \Rightarrow (Store\ relationships\ between\ the\ previous$

$identified\ ";"\ and\ this\ ";") \vee cont + 1))) \rhd$

GetEnrichedPathsWithLength The goal of this function is to get a list of enriched paths of a particular length ($length$) from rt. A list of paths is returned and "null" if no path of such length exists.

$GetEnrichedPathsWithLength(length, rt); out\ result : EnrichedPaths) \lhd$

$result = (\forall i < rt.paths \longrightarrow ((if(rt.paths[i].length = length) \Rightarrow true) \vee$

$false)) \rhd$

GetErtDivision The goal of this function is, given τ which is a set of forward and backward relationships at a particular position in a policy ($rels$), to process $rels$ separating and storing conditions of each direct relationship $fert$ and/or $bert$ in a list, as well as storing operators that join these relationships conditions in another list. Then, both lists are returned and "null" otherwise. Analogous to $GetPathsPolicies$, the order is a key matter and thus, returned lists have to be built keeping the order of $rels$.

$GetErtDivision(rels); out\ result : \{ListE, ListOpe\}) \lhd$

$result = (rels\ are\ processed\ storing\ rels.fert_i\ and\ rels.bert_i$

$in\ ListOfErtand\ operators\ that\ linked\ each\ rels.fert_i$

$and\ rels.bert_i\ in\ ListOfOperators)) \rhd$

GetFirstNode/GetLastNode The goal of these functions refers, respectively, to return the id of the first and last node of a particular path. A node id is returned if exists and "null" otherwise.

$GetFirstNode/GetLastNode(path); out\ result : STRING) \lhd$

$result = (return\ the\ first/\ last\ node) \rhd$

GetLengthPath The goal of this function is to identify the length of a given path ($path$). Then, given the operator ";" to differentiate hops in each path, the length $path$ is obtained by counting all semicolons plus one.

$GetLengthPath(path); out\ result : INTEGER) \lhd$

$cont = 1$

$result = (\forall i < path.length \longrightarrow (if(epath[i] = ";") \Rightarrow cont + 1)) \rhd$

GetNode The goal of this function is to identify the node in a path ($path$) located at a certain position (pos). A node id is returned and "null" otherwise.

$GetNode(path, pos); out\ result : STRING) \lhd$

$result = (return\ node\ located\ at\ pos) \rhd$

GetNumContacts This function returns the number of contacts of a given user (s).

$GetNumContacts(s\ result : INTEGER) \lhd$

$result = (s.contacts.length) \rhd$

GetPathsPolicies The goal of this function is, given a set of conditions that enriched paths in rt must satisfy (σ), to separate the conditions in predicates (ψ_i)

that different and independent enriched paths must satisfy, as well as creating a list of operators that join every ψ_i. Then, both lists are returned and "null" if an error occurs. Note that a key point is the order because the position of each operator refers to the enriched path to which it applies.

$$GetPathsPolicies(length, rt); out\ result : \{EnrichedPaths, ListOpe\}) \lhd$$
$$result = (\sigma\ is\ processed\ storing\ \sigma.\psi_i\ in\ EnrichedPaths\ and\ operators$$
$$that\ linked\ each\ \sigma.\psi_i\ in\ ListOpe\ \wedge MatchPathPolicy(\sigma.\psi_i, rt, 1)))\rhd$$

GetSubAtt/GetObjAtt These functions refer to "get" functions that return user attributes and object attributes respectively. They return attributes or "null" otherwise. Notice that input parameters depend on each function.

$$GetSubAtt/GetObjAtt(-; out\ result : S/O/RTATTRIBUTES) \lhd$$
$$result = (s/o/rt\ attributes)\rhd$$

Match The goal of this function is to verify the match between each value of a set of ω attributes ($ATT(\omega)$) and ω attributes involved in a particular policy (ρ_ω) [$VerifyDAttTypes/\ VerifyFVAttTypes/\ VerifyBAttTypes$]. It returns "true" if all attribute predicates are satisfied and "false" otherwise. Notice that ω corresponds only to the evaluation of a subject or an object.

$$Match(ATT(\omega), \rho_\omega; out\ result : BOOLEAN) \lhd$$
$$att(\omega)_i \in ATT(\omega)$$
$$result = (\forall att(\omega)_i \Rightarrow (if\ \ (att(\omega)_i.type = \mathcal{FV}) \Rightarrow$$
$$VerifyFVAttTypes(\gamma^j_{att(\omega)_i}, \rho_\omega)\ \vee\ (if(att(\omega)_i.type = \mathcal{D}) \Rightarrow$$
$$VerifyDAttTypes(\gamma^j_{att(\omega)_i}, \rho_\omega) \vee\ (if(att(\omega)_i.type = \mathcal{B}) \Rightarrow$$
$$VerifyBattTypes(\gamma^j_{att(\omega)_i}, \rho_\omega)))\rhd$$

MatchB/MatchC The goal of these functions focuses on verifying the satisfaction of conditions and obligations but they are very assorted and their implementation is left to systems' developers.

$MatchB/MatchC(s_1, o_1, r, \rho, \partial_b/\partial_c; out\ result : BOOLEAN) \lhd$
$result = (-) \rhd$

MatchDirectPaths The goal of this function is to verify that a set of direct forward and backward relationships found at a particular position of an enriched path of rt ($listPathsOfRT$) match those conditions τ of a policy ($pathsPolicy$) [$GetErtDivision$]. If conditions are met the result is "true", or "false" otherwise.

$MatchDirectPaths(listPathsOfRT, pathsPolicy); out\ result :$
$BOOLEAN) \lhd ListPOPolicy = GetErtDivision(pathsPolicy)$
$result = ((\forall i < ListPOPolicy.paths \longrightarrow (\forall j <\ listPathsOfRT \longrightarrow$
$(resultMatch =\ Match(listPathsOfRT[j], ListPOPolicy.paths[i]) \wedge$
$(if(resultMatch = true) \Rightarrow ListSatisfaction[i] = true))))$
$\wedge (\forall i < ListSatisfaction\ \longrightarrow ((if(ListSatisfaction[i] = true) \Rightarrow true)$
$\vee false))) \rhd$

MatchPathPolicy The goal of this function is to verify if an enriched path in rt matches a particular ψ which corresponds to the set of conditions of a path ($pathCond$). The general process consists of four steps. First, the length of the path required in $pathCond$ ($pLength$) [$GetLengthPath$] is calculated. Second, enriched paths of rt with the same length as $pLength$ are collected ($rtPathsL$) [$GetEnrichedPathsWithLength$]. Thirdly, paths in $rtPathsL$ are processed, getting the value of the attributes of the direct forward and backward relationships at every hop. Lastly, if the value of attributes of these paths match with those required in $pathCond$ [$MatchDirectPaths/\ GetDirectRelAtt$] is verified. Once the verification is completely and successfully performed, the result is "true" is a $pathCond$ is met, and "false" otherwise.

$MatchPathPolicy(pathCond, rt, \varpi); out\ result : BOOLEAN) \lhd$
$pLength = GetLengthPath(pathCond);$
$rtPathsL = GetEnrichedPathsWithLength(pLength, rt); cont = 0$

$result = (\forall i < rtPathsL \longrightarrow (\forall j < pLength \longrightarrow$
$(if(MatchDirectPaths(GetDirectRelAtt(rtPathsL[i], j),$
$GetDirectRelAtt(pathCond, j))) \Rightarrow ((cont + 1 \wedge (if(cont \geqslant \varpi) \Rightarrow true)) \vee$
$false))))\triangleright$

MatchR The goal of this function is to verify the match between the requested right $rReq$ and the right involved in an access control policy ($r\rho$). It returns "true" if rights match and "false" otherwise. Notice that ω corresponds to the evaluation of a subject or an object.

$MatchR(rReq, r\rho; out\ result : BOOLEAN) \triangleleft result = (rReq = r\rho)\triangleright$

MatchRT The goal of this function is to verify the satisfaction of ρ_{rt} given that rt is already built. The process consists of verifying the use of parameters ϖ and δ (in ρ_{rt}, that is $\rho_{rt}.\varpi$ and $\rho_{rt}.\delta$) and performing verifications accordingly. First, if $\rho_{rt}.\delta$ is not $\varnothing$, the existence of a clique is required and it has to be verified [$VerifyClique$]. Second, if $\rho_{rt}.\varpi$ is not $\varnothing$, it is studied if there exist a number $\rho_{rt}.\varpi$ of enriched paths satisfying all of them the conditions established the predicate ψ specified in $\rho_{rt}.\delta$ [$MatchPathPolicy$]. Finally, if $\rho_{rt}.\delta$ and $\rho_{rt}.\varpi$ are $\varnothing$, it is analysed if there exist a set of enriched paths satisfying the conditions established in the predicate . $\rho_{rt}.\sigma$ [$GetPathsPolicies/\ MatchPathPolicy/\ VerifyPolicy$]. Once the verification is successfully performed, the result is "true", and "false" otherwise.

$MatchRT(\rho_{rt}, rt; out\ result : BOOLEAN) \triangleleft$
$result = ((if(\rho_{rt}.\sigma = \varnothing \wedge \rho_{rt}.\varpi = \varnothing \wedge \rho_{rt}.\delta = \varnothing) \Rightarrow true) \wedge$
$(if(\rho_{rt}.\delta\ NOT\ \varnothing) \Rightarrow VerifyClique(\rho_{rt}, rt)) \wedge (if(\rho_{rt}.\delta\ NOT\ \varnothing) \Rightarrow$
$((if(MatchPathPolicy(\rho_{rt}.\sigma, rt, \rho_{rt}.\varpi))$
$\Rightarrow true) \vee false)) \wedge (if((\rho_{rt}.\delta\ NOT\ \varnothing) \wedge (\rho_{rt}.\varpi\ NOT\ \varnothing)$
$\Rightarrow (pathsDivided = GetPathsPolicies(\rho_{rt}.\sigma) \wedge (\forall i < pathsDivided.paths$
$\longrightarrow (pathsSatisfaction = MatchPathPolicy(pathsDivided.paths[i], rt, \rho_{rt}.\varpi)))$

$\wedge ((if(VerifyPolicy(pathsSatisfaction, pathsDivided.listOp))$
$\Rightarrow true) \vee false)))))\triangleright$

VerifyBAttTypes This function verifies the satisfaction of a particular attribute value ($\gamma^{j}_{att_i}$), being the attribute type $\mathcal{B}$, regarding particular policy predicates (*policyAttPred*). If conditions are met it returns "true", or "false" otherwise.

$VerifyBAttTypes(\gamma^{j}_{att_i}, policyAttPred; out\ result : BOOLEAN) \triangleleft$
$result = (\gamma^{j}_{att_i}\ is\ verified\ against\ policyAttPred\ )\triangleright$

VerifyClique The goal of this function is to verify the existence of a clique. The process involves a set of four steps. First, the number of paths of each particular length (*listlengthEP*) that are involved in a clique of a certain number of users (δ) [*CalculateCliquePaths*] is calculated. Second, *rt* is analysed, storing all paths whose length matches those stored in *listlengthEP* (*pathsClique*) [*pathsDivided/ GetEnrichedPathsWithLength*]. Thirdly, *pathsClique* is processed to verify enriched paths whose direct forward and backward relationships match ρ_{rt} (σ). If they match, they are stored (*acceptedPaths*) [*MatchDirectPaths/ GetDirectRelAtt/ GetFirstNode/ GetLastNode*]. Lastly, the result is "true" if nodes involved in *acceptedPaths* do not exceed δ [*GetNode*] and "false" otherwise.

$VerifyClique(rt, \delta, \sigma); out\ result : BOOLEAN) \triangleleft$
$listlengthEP = CalculateCliquePaths(\delta);$
$pathsDivided = GetErtDivision(\sigma)$
$result = ((\forall i < listlengthEP \longrightarrow (rtpaths[i] =$
$GetEnrichedPathsWithLength(i, rt) \wedge$
$((if(rtpaths[i].length \geqslant listlengthEP[i]) \Rightarrow$
$(pathsClique.ADD(rtpaths[i]))) \vee false))) \wedge$
$(\forall i < pathsClique.length \longrightarrow (\forall j < pathsClique[i].length$
$\longrightarrow ((if(MatchDirectPaths(GetDirectRelAtt($
$pathsClique[i], j), pathsDivided) \Rightarrow$

$acceptedPaths.ADD(pathsClique[i])) \vee false))) \wedge$

$listNodes.ADD(GetFirstNode(rt[i][j])) \wedge$

$listNodes.ADD(GetLastNode(rt[i][j])) \wedge (\forall i <$

$acceptedPaths.length \longrightarrow (\forall j < acceptedPaths[i].length$

$\longrightarrow (node = GetNode(acceptedPaths[i], j) \wedge$

$(if(node\ NOT\_IN\ listNodes) \Rightarrow listNodes.ADD(node)))))$

$\wedge ((if(ListNodes.length = \delta) \Rightarrow true) \vee false))\triangleright$

VerifyDAttTypes This function verifies the satisfaction of a particular attribute value ($\gamma^j_{att_i}$), being the attribute type $\mathcal{D}$, regarding particular policy predicates (*policyAttPred*). If conditions are met it returns "true", or "false" otherwise.

$VerifyDAttTypes(\gamma^j_{att_i}, policyAttPred; out\ result : BOOLEAN) \triangleleft$

$result = (\gamma^j_{att_i}\ is\ verified\ against\ policyAttPred\ given\ all\ applied\ \mathcal{X}\ operators)\triangleright$

VerifyFVAttTypes This function verifies the satisfaction of a particular attribute value ($\gamma^j_{att_i}$), being the attribute type $\mathcal{FV}$, i.e., $\mathcal{M}$ or $\mathcal{S}$, regarding particular policy predicates (*policyAttPred*). If conditions are met it returns "true", or "false" otherwise.

$VerifyFVAttTypes(\gamma^j_{att_i}, policyAttPred; out\ result : BOOLEAN) \triangleleft$

$result = (\gamma^j_{att_i}\ is\ verified\ against\ policyAttPred\ given\ all\ applied$

$\mathcal{F}\ and\ \mathcal{L}\ operators)\triangleright$

VerifyPolicy The goal of this function is, given a list of the result of evaluating each path ψ involved in σ (list of boolean values, *listPathsEval*) and a list of operators ($\wedge$ and/ or $\vee$) that connect each ψ in σ (*listOpe*), it is evaluated that *listPathsEval* matches with operators in *listOpe*. It should be noticed that elements of both list have to be evaluated against elements in the same position. Once the verification is completely performed the result is "true" if conditions are met, or "false" otherwise.

$VerifyPolicy(listPathsEval, listOpe); out\ result : BOOLEAN) \lhd$

$result = (Verify\ listPathsEval against\ listOpe) \rhd$

D.2 Enforcement functions for the extension of $SoNeUCON_{ABC}$

The access control enforcement functions applied to the extension of $SoNeUCON_{ABC}$ are the following:

FindCoOwners This function returns the list of the identifiers of co-owners of a given object o.

$FindCoOwners(o; out\ result : LIST\_OF\_STRING) \lhd$

$result = (co - owners\ of\ o) \rhd$

FindObjects This function returns all object parts o^j identifiers of an object o attached to a given user s.

$FindObjects(o, s; out\ result : LIST\_OF\_STRING) \lhd$

$result = (o^j\ of\ o\ linked\ to\ s) \rhd$

ProcessObject This function processes the requested object (o) according to objects parts ($listOij$) and evaluated access control policies ($listPoliciesResult$). Then, the objects is processed hidden objects parts accordingly.

$ProcessObject(o, listOij, listPoliciesResult; out\ result : LIST\_OF\_STRING) \lhd$

$result = (\forall listPoliciesResult \longrightarrow (if(listPoliciesResult[listOij_k] = true) \Rightarrow listOij_k\ grant\ access) \wedge (if(listPoliciesResult[listOij_k] = true) \Rightarrow listOij_k\ hidden)) \rhd$

UNIVERSIDAD CARLOS III DE MADRID

RESUMEN DE TESIS DOCTORAL

A user-managed access control model and mechanisms for Web Based Social Networks. Enhancing expressive power, co-ownership management, interoperability and authorized data exposures.

Autor:
Lorena González Manzano

Directores:
Dra. Ana Isabel González-Tablas Ferreres
Dr. Arturo Ribagorda Garnacho

DEPARTAMENTO DE INFORMÁTICA

Leganés, Junio de 2014

Índice

1. Introducción **3**

2. Motivación **5**

3. Metodología de investigación **6**

4. Contribuciones **9**

5. Conclusiones **14**

6. Análisis crítico **16**

7. Trabajos futuros **18**

1. Introducción

Nos encontramos en la era de la dependencia a la tecnología y a la hiper−conectividad [1] y las Redes Sociales (RSs) son grandes desarrollos en este contexto. La cantidad de RSs está aumentando en gran medida [2, 3] y desde el comienzo de Friendster en 2002, considerada la primera RS, hasta la actualidad, muchas han sido las RSs que se han desarollado, por ejemplo Facebook, MySpace, etc. Sin embargo, a pesar de los incuestionables beneficios que proporcionan estas aplicaciones, ej. la comunicación a lo largo y ancho del mundo, la seguridad y en concreto la privacidad, son importantes retos a los que hay que enfrentarse.

La privacidad se define como "*la condición de no tener conocimiento de información personal sobre aquello que es poseido por otras personas (1983)*"[4]. Las RSs almacenan gran cantidad de datos, muchos de ellos personales y estos deben ser cuidadosamente protegidos y gestionados, con independencia de que sea un tema de poco interés para los usuarios de las RSs [5] y a veces, incluso confuso [6]. En esencial, se consideran dos puntos de vista, el de los usuarios y el de los investigadores. Aunque los usuarios no consideran la privacidad como un requisito primordial, los investigadores promueven la creación de sistemas que, además de satisfacer las expectativas de los usuarios, protejan su seguridad.

En relación con la perspectiva de los usuarios, muchos autores han contribuido en este análisis. Becker *et al.* concluyó que el total de los usuarios de Facebook nunca habían usado ninguno de los mecanismos proporcionados para proteger la privacidad [7]. Asimismo, Acquisti *et al.* analizó que incluso los usuarios de Facebook que eran conscientes de los problemas de privacidad existentes continúan utilizándola.[5]. Este hecho puede relacionarse con la gran cantidad de beneficios que aparentemente proporcionan las RSs, así como por el hecho de que los usuarios pueden ser conscientes sobre la seguridad de Internet pero no de las amenazas a las que están expuestos [8]. Por el contrario, más recientemente, algunos estudios revelan que los usuarios se están preocupando cada vez más por la privacidad y por ello, cada vez relevan menos datos [9, 6].

Independientemente de los intereses y de las motivaciones de los usuarios de las RSs, estudiado por los investigadores y señalado por las autoridades, la privacidad es relevante en la vida de todas las personas. Por ejemplo, en la Declaración de los Derechos Humanos, el artículo 12 establece el derecho a no tener injerencias arbitrarias en la vida privada de ninguna persona [1].

[1] http://www.un.org/en/index.shtml , last access Feb. 2014

Del mismo modo, en base a la Directiva 95/46/EC del Parlamento Europeo del 24 de Octubre de 1995, el artículo 8 establece "*Los Estados miembros prohibirán el tratamiento de datos personales que revelen el origen racial o étnico, las opiniones políticas, las convicciones religiosas o filosóficas, la pertenencia a sindicatos, así como el tratamiento de los datos relativos a la salud o a la sexualidad*"[10]. A este respecto se han aplicado distintas técnicas y mecanismos para permitir a los usuarios controlar el acceso sobre sus datos. De hecho, la característica "ser privado"puede convertirse en una ilusión que ciega a los usuarios y les previene de identificar qué datos están realmente disponibles al público y qué datos están realmente protegidos [6]. En consecuencia, investigadores y profesionales trabajan en el desarrollo de medidas para la protección de los datos, pretendiendo conseguir el mismo nivel de privacidad que se puede encontrar fuera de la red [11].

Considerando la importancia y la necesidad de proteger la privacidad, junto con el aumento de las RSs, surge la siguiente pregunta: ¿Las RSs proporcionan los suficientes mecanismos para preservar la privacidad? Aunque se han realizado numerosos desarrollos, como son las herramientas que permiten aceptar o rechazar estar etiquetado en una foto, el control de acceso es un punto clave de investigación donde esta tesis contribuye.

Definido por el NIST "*el control de acceso se basa en determinar las actividades que son permitadas para los usuarios legítimos, analizando cada intento de acceso de un usuario para acceder a un recurso del sistema*" [12]. En el campo de las RSs se distinguen dos cuestiones asociadas con el control de acceso, una de ellas en relación con la protección frente a usuarios de las RSs [13] y otra de ellas relacionada con la protección frente a los proveedores de servicio de las RSs [14].

Por una parte, los desarrollos de control de accesos más comunes se basan en la creación de medidas de seguridad que permiten a los usuarios especificar quién accede a sus datos. Por ejemplo, si a unos determinados recursos, como son las fotos, se restringe el acceso a los amigos, intentos de acceso por parte de amigos de amigos deberían denegarse. Sin embargo, lo esencial es proporcionar control de acceso con alta granularidad, permitiendo que los usuarios puedan expresar todas sus preferencias con todo tipo de detalle, es decir, el gran reto es conseguir gestionar el control de acceso de forma expresiva [15]. Por ejemplo, las fotos relacionadas con el tema "Fiesta de verano"podrían estar disponibles para amigos desde Junio a Septiembre de 2014 y restringidas para amigos que también se consideren familiares. Otra cuestión a considerar es que, según modelos tradicionales, el control de acceso se gestiona con anterioridad a la entrega del dato. Por el contrario, algunos desarrollos más

actuales demandan nuevos requisitos que requieren gestionar el control de acceso a lo largo de todo el proceso de uso [16]. Por ejemplo, cuando las fotos tituladas "Fiesta" son accedidas, su descarga podría ser rechazada.

Además, dado que las RSs gestionan datos de gran cantidad de usuarios y muchos de estos datos pueden pertenecer a más de usuario distinto del propietario (que es el usuario que sube el dato a la RS), la gestión de la copropiedad es otro tema a considerar. Por ejemplo, en una foto de una banda de música pueden aparecer múltiples personas además del propietario de la misma, las cuales se convierten en copropietarias. Por tanto, el control acceso debe realizarse en base a las preferencias de todos los usuarios, propietarios y copropietarios.

Asimismo, considerando la gran cantidad y variedad de RSs, con distintos propósitos pero utilizando datos similares, la gestión del control de acceso es un proceso muy costoso [17]. Por ejemplo, el objetivo principal de Facebook es compartir fotos y comentarios entre amigos y amigos de amigos. De forma similar, pero directamente enfocado en conocer gente, Badoo permite compartir fotos y hacer comentarios. Los usuarios tienen que subir y gestionar el control de acceso en todas las RSs en las que tienen una cuenta, con independencia de que se utilicen los mismos datos en muchas de ellas.

Por otro lado, los proveedores de servicio de las RSs pueden ser una amenaza de la que hay que protegerse [14]. En la mayoría de las RSs los usuarios establecen qué usuarios pueden acceder a sus datos. Sin embargo, los proveedores de servicio tienen el control sobre todo aquello que tienen almacenado, pudiendo hacer uso de ello para sus propios intereses sin, en muchos casos, contar con la opinión y/ o el consentimiento de los usuarios.

2. Motivación

En general, el objetivo de la tesis es la mejora del control de acceso en las RSs, haciendo especial énfasis en permitir que los usuarios puedan expresar sus preferencias con alta granularidad. La cuestión es imitar la vida real [18]. Los investigadores de las RSs tienen que imitar las interacciones y comportamientos que las personas tienen en su vida cotidiana. En relación con esto, se debe facilitar la interacción entre distintos usuarios independientemente del tipo y propósito de la RS.

No obstante, debe recordarse que la privacidad ha de estar por encima de todo y además de imitar la vida real, la preservación de la privacidad es un requisito esencial [19].

3. Metodología de investigación

La metodología de investigación que se ha seguido para el desarrollo de esta tesis se describe en esta sección.

En primer lugar se ha llevado a cabo un profundo estudio del estado del arte en relación a la materia y objeto de estudio de la tesis. Se han estudiado las diferentes áreas relacionadas con el control de acceso y en concreto, con el control de acceso en RSs

El análisis del estado del arte ha permitido detectar los problemas o carencias y por tanto estipular los objetivos de investigación para esta tesis. En concreto, en esta tesis se abordan los siguientes cuatro problemas:

P1. Falta de granularidad en los sistemas de control de acceso para conseguir que los usuarios puedan controlar completamente sus datos.

Las RSs consisten en una gran cantidad de usuarios que disponen de gran cantidad de datos y que interactúan entre ellos mediante el establecimiento de relaciones. En consecuencia, hay que proporcionar los procedimientos adecuados que permitan a los usuarios la gestión de sus datos satisfaciendo todas sus necesidades y, a su vez, preservando su privacidad. Desde el desarrollo de los modelos de control de accesos tradicionales, generalmente asociados con los modelos de control de acceso Mandatorio, Discrecional y basado en Roles (MAC, DAC y RBAC respectivamente), muchos modelos se han desarrollado en el contexto de las RSs. Algunos de ellos se basan en la mejora o el refinamiento de los modelos tradicionales, como RBAC, para adaptarlos a las demandas de las RSs [20, 21]. Por otra parte, se han desarrollado nuevos modelos de control de acceso para satisfacer las necesidades impuestas por las RSs [22, 23, 24, 25]. Además, los modelos de control de uso son desarrollos a subrayar. Estos modelos se basan en la gestión del acceso con anterioridad o mientras se está haciendo uso del dato solicitado [26, 27]. El persistente control de los datos es una característica deseable en las RSs y los modelos de control de uso son destacables en este respecto. No obstante, los mecanismos y modelos de control de acceso para las RSs tienen un par de deficiencias. Primero, no son lo suficientemente expresivos como para permitir que los usuarios especifiquen todas sus preferencias y por ello, no se proporciona granularidad en la gestión del control de acceso, ej. la especificación de una relación con una determinada duración y la necesidad de tener un par de contactos comunes. Segundo, los modelos de control de acceso en las RSs no están enfocados en el control de uso.

P2. Falta de mecanismos para la gestión de la copropiedad que

satisfagan las preferencias de todos los usuarios sin restricciones.

Habitualmente los datos subidos a las RSs pertenecen a múltiples usuarios, en particular, al propietario, que es el usuario que carga los datos en las RSs, y a los copropietarios, que son los usuarios relacionados con los datos cargados. Por ello, el control de acceso tiene que considerar la gestión de las preferencias de ambos tipos de usuarios, propietarios y copropietarios. Asimismo, se han de satisfacer las preferencias de todos los usuarios para prevenir las violaciones de privacidad. Muchas propuestas utilizan mecanismos para gestionar la copropiedad basados en esquemas de votos en base a "lo que diga la mayoría"[28, 29]. Sin embargo, la privacidad de los usuarios que hayan escogido las preferencias más restrictivas se podría ver comprometida. En cambio, propuestas como la desarrollada por K. Thomas *et al.* [30] establecen que las políticas de control de acceso sólo se establecerán si se llega a un consenso entre todos los usuarios. Asímismo, esta solución es muy limitada porque, en muchos casos, los acuerdos pueden no llegar a encontrarse. Por tanto, la gestión de la copropiedad debe preservar la privacidad de los propietarios y de los copropietarios pero proporcionando flexibilidad en dicha gestión.

P3. Incapacidad para reutilizar y gestionar datos entre distintas RSs.

Existen gran varidad de RSs con múltiples propósitos y servicios. Por ejemplo, LinkedIn se enforca en el entorno profesional y Facebook está dirigida al público en general. Dada la falta de interoperabilidad, los usuarios tienen que crear muchas cuentas, una en cada RS en la que quieran participar. Por ello, la gestión de los datos puede convertirse en un proceso muy tedioso. Los datos se almacenan en cada una de las RSs en las que son cargados, ej. perfiles, fotos, etc., y deben gestionarse independientemente en cada RS. También hay que considerar que muchos de los datos utilizados en una RS son análogos a los utilizados en otra, no existiendo posibilidad de reutilización, ej. algunas de las fotos cargadas en Facebook pueden ser las mismas que las cargadas en MySpace. En relación con este problema se han desarrollado algunas propuestas. El estándar OpenID[2] es un ejemplo, el cual facilita la interoperabilidad de los datos de identidad. Otro ejemplo es el protocolo User−Managed Access (UMA) [31, 32]. Entre otras cuestiones, UMA proporciona a los usuarios el control sobre datos compartidos, lo cual es esencial para conseguir interoperabilidad en relación con los recursos y con las políticas de control de acceso. De hecho, en la práctica no existe ninguna propuesta que proporcione el acceso a los datos, ej. fotos, o la reutilización de los datos,

[2]http://openid.net/ , last access Feb. 2014

ej. políticas de control de acceso, desde una RS a otra.

P4. El desvelado de datos puede dar lugar a la violación de la privacidad de los usuarios.

Comúnmente conocido, muchas de las noticias que diariamente se nos presentan informan del continuo desvelado de datos almacenados en las RSs, bien develados por un proveedor de servicio[3] o bien por un atacante[4]. Los proveedores de las RSs disponen de los datos que los usuarios cargan en ellas y, una vez aceptados los "Términos de uso del servicio" en la fase de registro, los datos pueden ser utilizados lícitamente para distintos propósitos, ej. marketing. Por otro lado, los proveedores de servicio, además de utilizar los datos de forma lícita (lo cual puede llegar a violar la privacidad de los usuarios), indican la utilización de medidas de seguridad para prevenir el desvelado o la entrega de datos de forma desautorizada. Sin embargo, múltiples ataques se han producido en RSs muy populares[5], llegándose a comprometer los datos de los usuarios. Por ello, los datos tienen que ser protegidos contra atacantes internos o externos, es decir, contra el uso lícito de los datos por parte de los proveedores de servicio y contra el uso ilícito de los datos por parte de los atacantes. Independientemente de lo indicado por "Términos de uso del servicio", la privacidad debe preservarse. Para abordar este problema la criptografía ha sido una de las técnicas más utilizadas en la literatura [33, 34]. Los datos se almacenan cifrados y las claves de descifrado se distribuyen entre los usuarios autorizados. El problema principal de las RSs es que cada usuario está relacionado con un conjunto de usuarios con el que habría que distribuir las claves y, por ello, la gestión de las mismas es una tarea costosa a la que hay que buscar solución.

Considerendo los problemas anteriores, el objetivo de esta tesis es facilitar la gestión del acceso con alta granularidad, entre distintas RSs, a lo largo de todo el proceso de uso, y preservando la privacidad.

Por tanto, dada la necesidad de proporcionar dicho control, las necesidades a abordar en esta tesis se identifican en los siguientes objetivos:

O1. El desarrollo de un **modelo expresivo** que permita que los usuarios puedan expresar todas sus preferencias, asi como el **modelo administrativo asociado**.

[3]http://www.digitaleyemedia.com/blog/boxes−you−want−to−uncheck−on−linkedin, last access Feb. 2014

[4]http://www.facebook.com/notes/facebook−security/protecting−people−on−facebook /10151249208250766, last access Feb. 2014

[5]http://allfacebook.com/camera−bug−security−loophole−version−1−1−2_b107435, last access Feb. 2014

O2. El desarrollo de un **mecanismo para la gestión de la copropiedad** que permita **gestionar las preferencias de los propietarios y de los copropietarios**, satisfaciendo las preferencias de privacidad de todos ellos.

O3. El desarrollo de un **mecanismo para conseguir interoperabilidad y reusabilidad entre distintas RSs**, reduciendo la necesidad de tener distintas cuentas en distintas RSs.

O4. El desarrollo de un **mecanismo para proteger el acceso a los datos de forma no autorizada**, protegiéndose así la privacidad de los usuario y además, considerando una **fácil gestión de las claves utilizadas**.

4. Contribuciones

Las contibuciones de esta tesis son las siguientes:

C1. Modelo expresivo de control de uso para RSs, junto con el **modelo administrativo complementario**, que faciliten la gestión del control de acceso con alta granularidad. Dibujando a la RS como un grafo en el que los usuarios son los nodos y las relaciones son las aristas, se proponen $SoNeUCON_{ABC}$ y $SoNeUCON_{ADM}$. $SoNeUCON_{ABC}$ es una extensión del modelo $UCON_{ABC}$ que consigue expresividad en base a un total de seis características asociadas a las RSs. $UCON_{ABC}$ es un modelo de control de uso basado en la gestión de atributos. Este modelo es extendido para gestionar los atributos de los usuarios, de los datos y de las relaciones. La gestión del control de acceso se realiza preservando la privacidad, de modo que además de los atributos del administrador y del solicitante de un dato en particular, los atributos del resto de nodos asociados a la relación entre ambos, permanecen ocultos. Adicionalmente, se propone $SoNeUCON_{ADM}$, el modelo administrativo de $SoNeUCON_{ABC}$, el cual define la gestión de la revocación, la delegación y otras tareas administrativas. La adecuación de ambos modelos es evaluada. En $SoNeUCON_{ABC}$ se estudia, teóricamente, la posibilidad de gestionar todas las características identificadas. Además, el tiempo de ejecución de la verificación de las políticas es empíricamente analizado, concluyéndose así la posibilidad de implementar el modelo. Por otro lado, la evaluación de $SoNeUCON_{ADM}$ requiere analizar la completitud de las tareas administrativas proporcionadas.

C2. Mecanismo para la gestión de la copropiedad basado en **objetos que pueden descomponerse**, preservándose la privacidad de los propietarios y de los copropietarios. Los datos manejados en las RSs pueden no estar relacionados con un único usuario, sino con un conjunto de co-

propietarios, siendo éste el motivo por que el se ha desarrollado *Co−owned Personal Data management* (CooPeD). CooPeD presenta una novedosa técnica para preservar la privacidad de todos los usuarios, satisfaciendo todas las preferencias de los usuarios sin restricciones. Inspirado en [35], los objetos se descomponen en partes y a cada una de ellas se le asigna un usuario. Posteriormente, cada usuario, bien un propietario o un copropietario, gestiona el control de acceso sobre su parte considerando sus preferencias de privacidad. De igual modo, CooPeD se desarrolla sobre los modelos $SoNeUCON_{ABC}$ y $SoNeUCON_{ADM}$, extendiendo ambos para poder gestionar la copropiedad. CooPeD se evalúa analizando la posibilidad de implementarlo sobre $SoNeUCON_{ABC}$, desarrollando una implementación del mismo y realizando una encuesta para determinar su usabilidad.

C3. Mecanismos para conseguir interoperabilidad y reusabilidad entre distintas RSs incluyendo la **minimización sobre los datos expuestos**. Se han propuesto un par de protocolos desarrollados sobre una versión simplificada de $SoNeUCON_{ABC}$. Como primer paso, basado en el protocolo UMA [31] y en el proyecto FOAF [36], se presenta el protocolo para RSs UMA+FOAF (U+F). Este protocolo proporciona interoperabilidad en base a recursos, datos de identidad y políticas de control de acceso entre distintas RSs, donde los recursos se corresponden principalmente con las fotos, los videos o los comentarios, y los datos de identidad se corresponden con los perfiles y los datos de los contactos. Este protocolo gestiona relaciones directas. Seguidamente, se propone el protocolo extendido UMA+FOAF (eU+F), incluyendo la protección de los datos contra los proveedores de servicio de las RSs y la gestión de las relaciones indirectas, dado que esta última característica es esencial para gestionar el acceso con alta granularidad. Las relaciones indirectas facilitan la gestión de las características que las RSs deben gestionar, como son los múltiples caminos [13]. Esta contribución se ha evaluado calculando los tiempos de ejecución en las distintas fases de los protocolos, asi como realizando una comparación de los mismos con dos populares RSs, Facebook y MySpace. De igual modo, se compara conjuntamente U+F y eU+F.

En la Tabla 1, se puede identificar la relación entre los problemas identificados, los objetivos planteados y las contribuciones realizadas

Todas las cuestiones planteadas son, en nuestra opición, un paso para la mejora en la gestión del control de acceso en las RSs. Todas las contribuciones de esta tesis se resumen en la Figura 1. El proceso comienza cargando y gestionando los datos en las RSs, junto con el establecimiento de las políticas de control de acceso (mensaje 1 de la Figura 1). Posteriormente, los usuarios

Problema	**Objectivo**	**Contribución**
P1: Falta de granularidad en los sistemas de control de acceso.	O1: Modelo de uso expresivo y el administrativo asociado.	C1: Un modelo de uso expresivo para RSs y el complementario administrativo.
P2: Falta de mecanismos para la gestión de la copropiedad que satisfagan las preferencias de todos los usuarios.	O2: Mecanismo para gestionar la copropiedad satisfaciendo las preferencias de propietario y copropietarios.	C2: Un mecanismo para gestionar la copropiedad en objetos que pueden descomponerse.
P3: Incapacidad para reutilizar y gestionar datos entre distintas RSs.	O3: Interoperabilidad y reusabilidad entre distintas RSs.	C3: Mecanismo para alcanzar interoperabilidad y reusabilidad entre distintas RSs también minimizando el acceso a datos desautorizados.
P4: El desvelado de datos puede dar lugar a la violación de la privacidad.	O4: Mecanismo para proteger el acceso a los datos de forma no autorizada.	

Cuadro 1: Relación ente problemas, objetivos y contribuciones.

pueden solicitar datos de otros usuarios, bien estando dichos usuarios en su misma RS o en otra diferente, incluyendo usuarios directamente (mensaje 2 de Figura 1) o indirectamente (mensaje 3 de la Figura 1) conectados. Además, los accesos desautorizados a los datos se previenen aplicando criptografía y gestionando el control de acceso a lo largo de todo el proceso de uso (mensaje 3 de la Figura 1). Asimismo, hay que considerar que en caso de existir datos en copropiedad, el control de acceso se realiza en base a las políticas establecidas por los propietarios y por los copropietarios de los datos (mensaje 4 de la Figura 1)

Finalmente, hay que tener en cuenta que problemas de privacidad causados por el uso de elementos externos, como pueden ser las cámaras digitales, u otros sistemas para la grabación de las actividades que se realizan en la pantalla, están fuera del alcance de esta tesis. Por otro lado, se considera que los usuarios con los que se establece una relación siempre son considerados confiables. Por ello, la identificación de cambios inesperados en el comportamiento de los usuarios se deja para trabajo futuro. En caso de que el comportamiento de los usuarios difiera del común, ej. utilizando un servidor de anonimato para preservar su identidad, dichos usuarios se considerarían como no con-

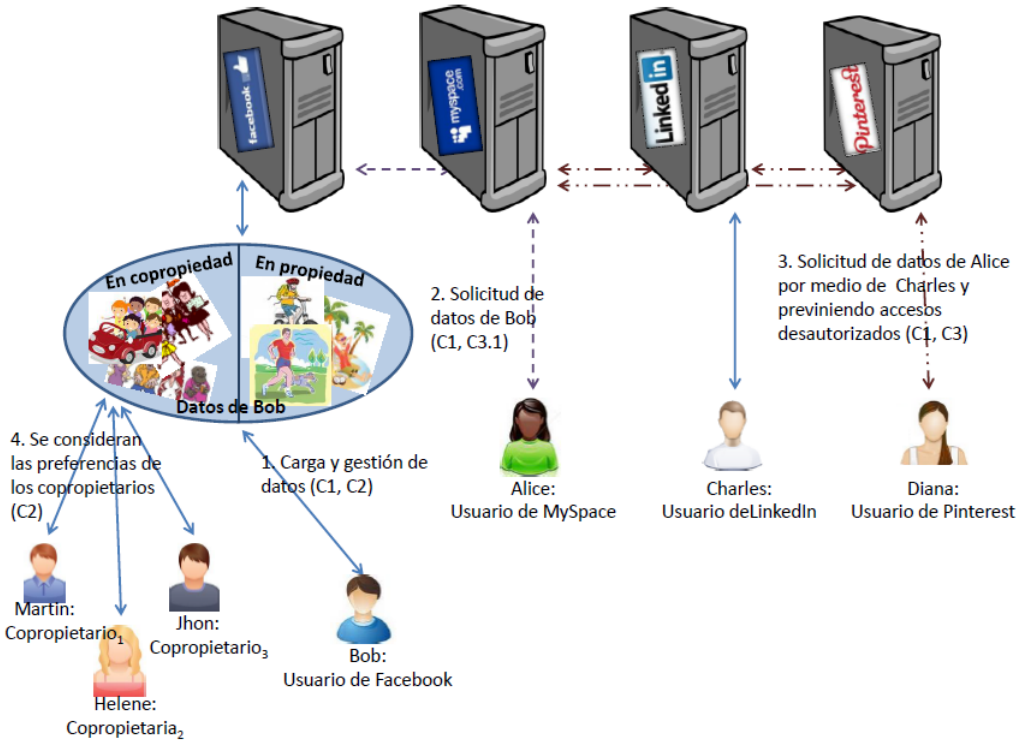


Figura 1: Visión general del las contribuciones

fiables, siendo las herramientas para la detección de estos comportamientos un primer paso en la investigación de este problema.

Los resultados de investigación publicados en revistas y congresos científicos durante el desarrollo de la tesis se listan a continuación:

Publicados (basados en la tesis):

1. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. “U+F Social Network Protocol: Achieving interoperability and reusability between Web Based Social Networks". 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, 2012.

2. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, B. Ramos. “Control de Acceso en Redes Sociales Web". XII Reunión española sobre Criptología y Seguridad de la Información (RECSI), 2012.

3. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. Security and Privacy Preserving in Social Networks. Springer.

Capíputulo “User-Managed Access Control in Web Based Social Networks", 2013.

4. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. "Seguridad en Redes Sociales: problemas, tendencias y retos futuros", VII Congreso Iberoamericano en Seguridad Informática (CIBSI), 2013.

5. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. “SoNeUCON$_{ABC}$, an expressive usage control model for Web-Based Social Networks", Computers & Security, 2014.

6. L. González-Manzano, B. Brost, M.Aumüller. “An architecture for trusted PaaS cloud computing for personal data", Workshop Wissenschaftliche Ergebnisse der Trusted Cloud Initiative, Springer Verlag, 2014.

7. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. “Extended U+F Social Network Protocol: Interoperability, reusability, data protection and indirect relationships in Web Based Social Networks", Systems and Software.

8. L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. “$SoNeUCON_{ADM}$: the administrative model for $SoNeUCON_{ABC}$ usage control model", XIII Reunión española sobre Criptología y Seguridad de la Información (RECSI), 2014.

   L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. “CooPeD: Co-owned Personal Data Management", Computers & Security, 2014.

Publicados (relacionados con la tesis):

9 L. González-Manzano, B. Brost, M.Aumüller. “An architecture for trusted PaaS cloud computing for personal data", Workshop Wissenschaftliche Ergebnisse der Trusted Cloud Initiative, Springer Verlag, 2014 in Press.

Enviados (relacionados con la tesis):

10 E. Palomar, L. González-Manzano, A. Alcaide, A. Galán. “Implementing a Privacy-enhanced ABC System for Online Social Networks with Co-Ownership Management", Computer Networks.

11 L. González-Manzano, Ana I. González-Tablas, J. M de Fuentes, A. Ribagorda. “Towards a trade-off between privacy and rewards in Web-Based Social Networks advertising", Information Systems.

12 L. González-Manzano, J. M de Fuentes, Ana I. González-Tablas, A. Ribagorda. “VAADapt - Adapting VAAD for an enriched and privacy-preserving advertisement dissemination in VANETs", Information Sciences.

5. Conclusiones

Esta tesis se basa en proporcionar control de acceso con alta granularidad entre distintas redes sociales, preservando la privacidad. Esto ayuda a disminuir las amenazas causadas por tres cuestiones. Primero, los sistemas de control de acceso impiden que los usuarios expresen todas sus preferencias y controlen la copropiedad de sus datos. Segundo, la gestión del control de acceso en distinta RSs es un proceso tedioso e incómodo. Finalmente, los proveedores de servicio de las RSs almacenan los datos y estos pueden utilizarse para propósitos no deseados.

Existen muchas contribuciones relacionadas con la gestión del control de acceso en RSs, asi como mecanismos que son candidatos para conseguir interoperabilidad y reusabilidad entre distintas RSs. Sin embargo, las propuestas existentes no están enfocadas en conseguir expresividad en la gestión del acceso, ni tampoco interoperabilidad, reusabilidad y prevención contra accesos desautorizados a los datos. Para abordar estos problemas, esta tesis propone un modelo de control de uso expresivo, asi como el modelo administrativo complementario. Sobre ambos modelos se desarrollan un mecanismo para la gestión de la copropiedad y un par de protocolos para conseguir interoperabilidad y reusabilidad entre distintas RSs.

La primera contribución de esta tesis consiste en la definición del modelo de control de uso anteriormente mencionado, asi como el modelo administrativo que le acompaña, denominados $SoNeUCON_{ABC}$ y $SoNeUCON_{ADM}$ respectivamente. $SoNeUCON_{ABC}$ es un modelo de control de uso expresivo, que permite gestionar un total de seis características (distancia, contactos comunes, cliqués, múltiples caminos, dirección y elementos flexibles en las políticas de control de acceso) a lo largo de todo el proceso de uso. Teóricamente se ha identificado que todas las características pueden ser gestionadas por el modelo y empíricamente, se ha analizado que el modelo puede implementarse en la mayoría de los casos. Para realizar la evaluación empírica se ha creado una prueba de concepto. Considerando que 2.000 ms. es el tiempo máximo que un usuario va a esperar para que la información se le muestre por pantalla, los resultados muestran que la verificancián de políticas que no incluyen cliqués es satisfactoria si la cantidad de nodos explorados no excede

de 200.000 y 200 relaciones por nodo. Además, la evaluación de políticas con cliqués se considera exitosa si se exploran menos de 30.000 nodos y cada nodo tiene menos de 200 relaciones. Por tanto, es posible considerar la adecuación del modelo en el contexto de las RSs.

Asimismo, $SoNeUCON_{ADM}$, el modelo administrativo de $SoNeUCON_{ABC}$, proporciona todas las tareas administrativas identificadas basadas en la gestión de los derechos de uso y de los derechos administrativos. La completitud del modelo se identifica mediante una comparación teórica con los modelos administrativos asociados a $UCON_{ABC}$. por ser el modelo en que se basa, y a RBAC, por disponer de un modelo administrativo muy maduro.

Intentando gestionar la copropiedad de los datos en las RSs y tomando como base los modelos $SoNeUCON_{ABC}$ y $SoNeUCON_{ADM}$, la segunda contribución de esta tesis es un mecanismo llamado *Co−owned Personal Data management*, CooPeD. Éste es un novedoso mecanismo basado en la gestión de objetos que pueden descomponerse, considerando las preferencias de los propietarios y de los copropietarios. Utilizando la misma prueba de concepto desarrollada para la evaluación de $SoNeUCON_{ABC}$, se ha estudiado la posibilidad de implementar CooPeD sobre dicho modelo. Como resultado se concluye que si un par de usuarios está relacionado con una estructure rt compuesta por relaciones directas, se pueden llegar a evaluar un máximo de 34 políticas por objeto sin exceder el umbral de los 2000 ms. Por el contrario, si un par de usuarios está relacionado con una estructura rt compuesta por relaciones indirectas (de 2 saltos), sólo es posible evaluar 4 políticas por objeto. Además, se ha desarrollado un prototipo para verificar la posibilidad de implementar CooPeD. Por último, y en base al prototipo anterior, se ha realizado una encuesta para estudiar la utilidad del mecanismo, la cual ha sido completada por 206 personas de todo el mundo, llegando a la conclusión de que el 72.6 % de los encuestados podrían llegar a ser potenciales usuarios de CooPeD.

Además de las cuestiones anteriores, la gran cantidad de RSs y la gestión de los datos en todas ellas puede provocar problemas de privacidad. Los usuarios tienen que gestionar los datos en todas las RSs en las que tengan una cuenta. Por ello, basado en una simplificación de $SoNeUCON_{ABC}$, la tercera contribución consiste en el desarrollo de un par de protocolos para conseguir interoperabilidad y reusabilidad entre distintas RSs. Uno de los protocolos, denominado protocolo UMA+FOAF (U+F), proporciona interoperabilidad y reusabilidad de datos de identidad, recursos y políticas de control de acceso entre distintas RSs, consiguiendose así simplificar la gestión del control de acceso. Extendiendo este protocolo, se propone el protocolo extendido

UMA+FOAF (eU+F) para la protección de los datos frente a los proveedores de servicio y la gestión de las relaciones indirectas. Ambos protocolos se evalúan empírica y teóricamente. La satisfacción de todos los requisitos establecidos se ha analizado de forma teórica, así como los efectos de la reutilización en las distintas fases del protocolo. Además, se ha creado un prototipo, compuesto por dos RSs, FriendBook+ y MyLeisure, para analizar la posibilidad de implementar y desplegar estos protocolos. Este estudio consiste en medir el tiempo de ejecución de cada una de las fases de los protocolos, junto con una comparación con dos RSs de alta popularidad, Facebook y MySpace. Los resultados demuestran que, en general, acceder al perfil y a una foto de un contacto en Facebook o en MySpace produce un menor tiempo de ejecución que acceder al perfil y a una foto desde FriendBook+ a MyLeisure. Sin embargo, éste tiempo es comparable con el prototipo cuando algunos elementos son reutilizados.

6. Análisis crítico

El modelo de control de uso $SoNeUCON_{ABC}$, basado en proporcionar control de acceso con alta expresividad en las RSs, sienta las base de todas las contribuciones de esta tesis. Sin embargo, la adecuación del lenguaje de políticas propuesto, en relación con las expectativas y los intereses de los usuarios, debe ser evaluado.

Una questión importante se plantea en cuanto al motivo por el cual construir $SoNeUCON_{ABC}$ a partir de $UCON_{ABC}$ en lugar de utilizar otro modelo. En un primer momento la necesidad de gestionar las relaciones apoya la utilización de modelos basados en relaciones (RelBAC). Por el contrario, aunque las relaciones son elementos fundamentales, los estudios revelan la necesidad de gestionar, entre otros, atributos de usuarios y de objetos. Por tanto, atendiendo a las demandas de las RSs, la gestión de los atributos es esencial, siendo $UCON_{ABC}$ un buen punto de partida.

En relación con el modelo administrativo, $SoNeUCON_{ADM}$, su implementación, bien en un entorno real o simulado, se plantea como trabajo futuro. Específicamente, la evaluación realizada no estudia la satisfacción de los usuarios.

CooPeD, el mecanismo propuesto para la gestión de la copropiedad, trabaja sobre objetos que pueden descomponerse pero su gestión presenta dos limitaciones. Por un lado, la descomposición de objetos requiere la identificación de las partes de cada uno de los objetos, siendo necesario superar las limitaciones de las herramientas para descomponer los objetos. Los propietarios

pueden descomponer manualmente los objetos pero las descomposiciones automáticas son preferibles. Por ejemplo, la descomposición de un conjunto de usuarios que están bailando y abrazándose entre ellos es una tarea de gran complejidad. Por otra parte, la gestión de partes de los objetos que pertenecen a múltiples usuarios ha de ser estudiada, aunque éste es un tema que está fuera del ámbito de esta tesis y se presenta como trabajo futuro. Por ejemplo, asumiendo que una pareja está enfrente de su coche, la imagen se descompone en tres partes, dos partes en relación con cada uno de los miembros de la pareja y otra asociada con el coche. En este escenario se debe determinar quién y cómo se han de establecer las políticas de control de acceso sobre el coche.

Respecto a conseguir interoperabilidad y reusabilidad entre distintas RSs se ha desarrollado U+F y su versión extendida eU+F. Ambos protocolos se construyen sobre una versión simplificada del modelo $SoNeUCON_{ABC}$. Sin embargo, se ha discutido cómo eU+F puede extenderse para gestionar todas las características del modelo $SoNeUCON_{ABC}$. En concreto, en relación con el control de uso, cuando se detectan modificaciones, inclusiones o borrados en los atributos o en las políticas, es posible que se requiera la re-evaluación de las políticas. No obstante, este hecho puede afectar al rendimiento del protocolo, siendo indispensable un riguroso análisis en base a la gestión del control de uso. Por otra parte, eU+F debe gestionar todas las características que $SoNeUCON_{ABC}$ proporciona, tanto en la gestión de la estructura rt como en la evaluación de políticas sobre dicha estructura. Sin embargo, la construcción de rt es compleja y se ha de estudiar el mejor modo de realizar su construcción.

En base a los modelos de confianza propuestos en los protocolos presentados, el propuesto en U+F es bastante restrictivo, siendo éste mejorado en eU+F. En primer lugar, eU+F asume proveedores de identidad (IdP) confiables. Entidades que almacenan datos personales son presumiblemente confiables ya que, de lo contrario, la privacidad de los usuarios podría violarse. En concreto, los IdPs almacenan datos de identidad y las claves de descifrado de los recursos, de modo que si estas entidades actúan maliciosamente todos los datos de los usuarios podrían estar comprometidos. De forma similar, otra suposición es considerar a los gestores de autorización (AMs) entidades confiables. Esta suposición también ayuda a proteger la privacidad de los usuarios. La información recibida por los AMs es utilizada para evaluar las políticas de control de acceso y conseguir los tokens. Sin embargo, los tokens tienen que ser firmados por la RS adecuada y presentados a los Hosts e IdPs correspondientes para, finalmente, obtener los datos solicitados. En una situación opuesta, esto es, en caso de que AMs no confiables entren en

juego, estos pueden entregar tokens a las RSs y éstas podrían acquirir datos ilegítimamente.

Finalmente, hay que analizar la gestión de los datos tras su entrega. Esta cuestión, conocida con el nombre de *sticky policies* en base a los requisitos de Gates et al. [37], es abordada en esta tesis mediante la construcción de $SoNeUCON_{ABC}$ a partir del modelo de control de uso $UCON_{ABC}$. Sin embargo, aunque este modelo gestiona el control de uso, todavía se considera trabajo futuro la realización práctica de este requisito.

7. Trabajos futuros

Las propuestas presentadas en esta tesis están abiertas a nuevos desarrollos, los cuales contribuyen en la completitud de este trabajo, así como proporcionan una amplia visión del control de acceso en RSs.

Respecto a $SoNeUCON_{ABC}$, sería deseable la implementación de un algoritmo eficiente para la evaluación de cliqués, así como un análisis a gran escala. Asimismo, la identificación de un catálogo de atributos (de objetos, de sujetos y de relaciones) se plantea como un trabajo deseable. Además, también se ha de estudiar la complejidad en la construción de las políticas.

Por otro lado, el siguiente paso en relación con la mejora de $SoNeUCON_{ADM}$ es la gestión temporal de las delegaciones.

El estudio de la curiosidad que los usuarios pueden tener al utilizar CooPeD es otra línea de trabajo futuro. Particularmente, aunque se utilicen técnicas de ocultación sofisticadas para ocultar las partes de los objetos, se han de estudiar cuestiones como ¿quién/ qué está bajo la parte escondida del objeto? o ¿cómo puedo llegar a conocer/lo?, etc.

Otra línea de desarrollo futuro es la mejora del prototipo implementado para CooPeD. Se deberían desarrollar técnicas sofisticadas para la detección de las siluetas de los usuarios, consiguiendo realizar ocultaciones con alta precisión. De igual modo, el prototipo podría extenderse para detectar, no sólo personas, sino también vehículos, animales, etc.

En relación con U+F y eU+F, dado que el segundo se basa en el primero los retos futuros se asocian a eU+F. Este protocolo puede extenderse de distintos modos. Primero y más importante, el prototipo desarrollado debe desplegarse en un entorno real, incluyendo la gestión de relaciones indirectas y el control de uso a lo largo de todo el proceso de uso. También, eU+F debe funcionar sobre $SoNeUCON_{ADM}$.

Además, la especificación de restricciones y reglas para determinar qué se considera un IdP, una RS o un AM (Gestor de Autorización, en inglés *Authorization Manager*) confiable, es una cuestión abierta. Otra cuestión relevante es que, actualmente, el protocolo aborta si un AM, IdP o RS no es confiable. Por tanto, como trabajo futuro se plantea la especificación dinámica de entidades confiables.

Asociado a todas las contribuciones de esta tesis, un siguiente paso de desarrollo es la protección de la privacidad de los usuarios previniendo a las RSs la inferencia de las relaciones de los usuarios. Actualmente, las relaciones sociales pueden inferirse y aunque esto no afecta directamente a la privacidad de los usuarios se han de establecer contramedidas frente a las necesidades demandadas. En concreto, propuestas como la desarrollada por Carminati et al. [38], basadas en la protección de las relaciones por medio de certificados, se pueden tomar como punto de partida.

También relacionado con todas las contribuciones, J. Park *et al.* propone la distinción entre usuarios y sesiones [39], pudiendo llegar a definir políticas en base a las sesiones de los distintos usuarios. Por ello, el trabajo futuro consiste en el estudio de nuevas propuestas para incluir esta cuestión en los modelos y los mecanismos propuestos.

Por último pero no menos importante, la especificación de técnicas y mecanismos de control de datos tras su entrega, es otra línea futura. En cierto modo esta característica se puede considerar satisfecha debido a la aplicación de un modelo de control de uso. Sin embargo, aunque hay trabajos realizados en esta dirección [40, 41] y esta tesis proporciona algunas guías, se requiren más detalles.

Referencias

[1] Harper, R. and Rodden, T. and Rogers, I. and Sellen, A. Being Human: Human-Computer Interaction in the Year 2020. Microsoft Corporation; 2008.

[2] Ellison, N. B. and others. Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication. 2007;13(1):210–230.

[3] Kumar, R. and Novak, J. and Tomkins, A. Structure and evolution of online social networks. In: Link Mining: Models, Algorithms, and Applicationsp.337–357.

[4] Parent, W. A. Privacy, Morality, and the Law. Philosophy and Public Affairs. 1983;12(4):269–288.

[5] Acquisti, A. and Gross, R. Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. In: Privacy Enhancing Technologies. Lecture Notes in Computer Sciencep.36–58.

[6] Hoadley, C. M. and Xu, H. and Lee, J. J. and Rosson, M. B. Privacy as information access and illusory control: The case of the Facebook News Feed privacy outcry. Electronic Commerce Research and Applications. 2010;9(1):50 – 60.

[7] Becker, J. and Chen, H. Measuring Privacy Risk in Online Social Networks. In: Proceedings of W2SP 2009: Web 2.0 Security and Privacy; 2009. .

[8] Oracle-Team. Online security, A Human Perspective. 2011;.

[9] Dey, R. and Jelveh, Z. and Ross, K. W. Facebook Users Have Become Much More Private: A Large-Scale Study. In: Proceedings of SESOC 2012; 2012. .

[10] European Parliament. Directive 95/46/EC of the European Parliament and of the Council; 1995.

[11] Dwyer, C. and Hiltz, S. R. and Passerini, K. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In: Proceedings of AMCIS; 2007. .

[12] Hu, V. C. and Ferraiolo, D. and Kuhn, D. R. Assessment of access control systems. US Department of Commerce, National Institute of Standards and Technology; 2006.

[13] Carminati, B. and Ferrari, E. Access control and privacy in web-based social networks. In: International Journal of Web Information Systems. 4p.395–415.

[14] Foresti, S. Preserving privacy in data outsourcing. Springer; 2010.

[15] Fong, P. W. L. and Siahaan, I. Relationship-based access control policies and their policy languages. In: Proceedings of the 16th ACM symposium on Access control models and technologies. SACMAT '11. ACMp.51–60.

[16] Lazouski, A. and Martinelli, F. and Mori, P. Usage control in computer security: A survey. Computer Science Review. 2010;4(2):81 – 99.

[17] Anderson, J. and Stajano, F. Not That Kind of Friend: Misleading Divergences Between Online Social Networks and Real-World Social Protocols. In: Proceedings of the Seventeenth International Workshop on Security Protocols (SPW' 09). Citeseerp.1–6.

[18] PrimeLife-members. Requirements and concepts for privacy-enhancing access control in social networks and collaborative workspaces. 2009;.

[19] Lazer, D. and Pentland, Alex S. and Adamic, L. and Aral, S. and Barabasi, A. L. and others. Life in the network: the coming age of computational social science. Science (New York, NY). 2009;323(5915):721.

[20] Li, J. and Tang, Y. and Mao, C. and Lai, H. and Zhu, J. Role Based Access Control for social network sites. In: Pervasive Computing (JCPC), 2009 Joint Conferences onp.389 –394.

[21] Ding, J and Mo, L. Enforcement of Role Based Access Control in Social Network Environments. In: Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference onp.92 –101.

[22] Giunchiglia, F. and Zhang, R. and Crispo, B. RelBAC: Relation Based Access Control. In: Semantics, Knowledge and Grid, 2008. SKG '08. Fourth International Conference onp.3 –11.

[23] Fong, Philip W. L. Relationship-based access control: protection model and policy language. In: Proceedings of the first ACM conference on Data and application security and privacy. CODASPY '11. ACMp.191–202.

[24] Carminati, B. and Ferrari, E. and Perego, A. Rule-Based access control for social networks. In: Proceedings of the 2006 international conference on On the Move to Meaningful Internet Systems: AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET - Volume Part II. OTM'06. Springer-Verlagp.1734–1744.

[25] Masoumzadeh, A. and Joshi, J. OSNAC: An Ontology-based Access Control Model for Social Networking Systems. In: Social Computing (SocialCom), 2010 IEEE Second International Conference onp.751 –759.

[26] Park, J. and Sandhu, R. and Cheng, Y. ACON: Activity-Centric Access Control for Social Computing. In: Availability, Reliability and Security (ARES), 2011 Sixth International Conference onp.242 –247.

[27] Kumari, P. and Pretschner, A. and Peschla, J. and Kuhn, J-M. Distributed data usage control for web applications: a social network implementation. In: Proceedings of the first ACM conference on Data and application security and privacy. ACMp.85–96.

[28] Hu, H. and Ahn, G-J. and Jorgensen, J. Multiparty Access Control for Online Social Networks: Model and Mechanisms. IEEE Transactions on Knowledge and Data Engineering. 2012;99.

[29] Squicciarini, A. C. and Xu, H. and Zhang, X. L. CoPE: Enabling collaborative privacy management in online social networks. Journal of the American Society for Information Science and Technology. 2011;62(3):521–534.

[30] Thomas, K. and Grier, C. and Nicol, D. M. unfriendly: multi-party privacy risks in social networks. In: Proceedings of the 10th international conference on Privacy enhancing technologies. PETS'10p.236–252.

[31] Machulak, M. P. and Maler, E. L. and Catalano, D. and van Moorsel, A. User-managed access to web resources. In: Proceedings of the 6th ACM workshop on Digital identity management. DIM '10p.35–44.

[32] T. Hardjono, Ed. User-Managed Access (UMA) Core Protocol, draft-hardjono-oauth-umacore-05C; 2012. Available from: `http://docs.kantarainitiative.org/uma/draft-uma-core.html`, `lastaccessApr.2014`.

[33] Graffi, K. and Groß, C. and Stingl, D. and Hartung, D. and Kovacevic, A. and Steinmetz, R. LifeSocial.KOM: A Secure and P2P-based Solution

for Online Social Networks. In: Proceedings of the IEEE Consumer Communications and Networking Conference. IEEE Computer Society Press; 2011. .

[34] Lucas, M. M. and Borisov, N. FlyByNight: mitigating the privacy risks of social networking. In: Proceedings of the 7th ACM Wks. on Privacy in the electronic society. WPES '08. ACMp.1–8.

[35] Zhu, F. and Lv, Q. ACEAC: A Novel Access Control Model for Cooperative Editing with Workflow. In: Electronic Commerce and Security, 2008 International Symposium onp.1010 –1014.

[36] FOAF Team. *FOAF* project; 2000. . `http://www.foaf-project.org/,lastaccessApr.2014`.

[37] Gates, C. Access control requirements for web 2.0 security and privacy. IEEE Web. 2007;2(0).

[38] Carminati, B. and Ferrari, E. and Perego, A. Private Relationships in Social Networks. In: Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Wks. IEEE Computer Societyp.163–171.

[39] Park, J. and Sandhu, R. and Cheng, Y. A User−Activity−Centric Framework for Access Control in Online Social Networks. Internet Computing, IEEE. 2011;15(5):62 –65.

[40] Squicciarini, A. C. and Sundareswaran, S. Web-traveler policies for images on social networks. World wide web. 2009;12(4):461–484.

[41] Jahid, S. and Nilizadeh, S. and Mittal, P. and Borisov, N. and Kapadia, A. DECENT: A decentralized architecture for enforcing privacy in online social networks. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on. IEEEp.326–332.